DEC-11-ORSMA-B-D

RSTS-11
System Manager's Guide

FOR USE WITH RSTS-11

Version V004A

PDP-11 RESOURCE TIME-SHARING SYSTEM

January 1973

FOR ADDITIONAL COPIES, ORDER NO. DEC-11-ORSMA-B-D FROM DIGITAL EQUIPMENT CORPORATION, SOFTWARE DISTRIBUTION CENTER, MAYNARD MASSACHUSETTS 01754

digital equipment corporation · maynard. massachusetts

First Printing, May, 1971
Revised, October 1972
Revised, January 1973

Your attention is invited to the last two pages of this document. The "How-to-Obtain Software Information" page tells you how to keep up-to-date with DEC's software. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and are considered when documenting subsequent manuals.

Technical changes are noted in this manual by change bars in the outside margin of the page or by dated pages.

Associated Documents:

<u>BASIC-PLUS Language Manual</u>, DEC-11-ORBPA-A-D

<u>RSTS-11 System User's Guide</u>, DEC-11-ORSUA-A-D

CONTENTS

PREFACE

This addendum to the RSTS-11 System Manager's Guide, order number DEC-11-ORSMA-B-DN2, describes the system generation procedures for version VØ4B-17 of RSTS-11. Also included is information concerning a new initialization option (PATCH) and new system programs and SYS system function calls. All known deficiencies in Version 4A-12 have been corrected in Version 4B-17.

To update the subject manual, perform the following steps.

Replace the old Preface and CONTENTS pages with new CONTENTS pages.

Replace the old Chapter 2 with the new Chapter 2.

Replace pages 3-7, 3-8, 3-9, and 3-10 with new pages 3-7, 3-8, 3-9, 3-9.1, 3-9.2, 3-9.3 and 3-10.

Replace old pages 4-17 and 4-18 with new pages 4-17, 4-18, 4-18.1 and 4-18.2.

Replace old pages 5-1 through 5-8 with new pages 5-1 through 5-8.1.

Replace old page 5-39 with new pages 5-39 through 5-49.

Remove Chapter 7.

Replace old Appendix A with new Appendix A.

Replace old Appendix C with new Appendix C.

Replace old Appendix E with new Appendix E.

Remove Appendix G.

Replace the old HOW TO OBTAIN SOFTWARE INFORMATION page with the new one.

Replace the old READER'S COMMENTS page with the Addendum READER'S COMMENTS page.

Place the Addendum Title page behind the current title page.

Place this Preface page behind the new CONTENTS pages and annotate the current CONTENTS page with the following: Updated by Addendum DEC-11-ORSMA-B-DN2 dated February 1975.

CHAPTER 1

INTRODUCTION TO RSTS-11 SYSTEM MANAGEMENT

1.1 OVERVIEW OF SYSTEM STRUCTURE

The RSTS-11 Resource Time Sharing System is comprised of PDP-11 hardware units and system software components which allow multiple users simultaneous, time-shared access to the full computational and data-processing power of the BASIC-PLUS language, to an efficient and versatile mass-storage structure, and to a shared set of chosen peripheral resources.

1.1.1 System Hardware

The heart of the RSTS-11 hardware is the PDP-11 Central Processor Unit (CPU), equipped with a time-clock. The interrupt, stack, general-registers, and Unibus structures ideally equip the PDP-11 CPU for governing and controlling the time-shared processing.

A system disk (RF11, RK11, or possibly an RK11 in combination with an RC11) provides the necessary mass-storage for the system, and can be augmented by additional disk structures.

Local terminals and dataphone ports to remote terminals open the system interactively to as many as 16 simultaneous users. These terminals may be standard Teletypes[1], DECwriters, or video-type terminals (VTØ5). Moreover, when these terminals are not logged into the system they are available to other users as programmable peripheral devices.

Finally, the power of the RSTS-11 system hardware can be complemented by a full set of shared peripheral resources, such as line printer, card reader, magnetic tape, DECtape, and high-speed paper tape.

---

[1] Teletype is a registered trademark of the Teletype Corporation.

A typical small system, capable of being expanded by the addition of Teletypes or other terminals, is as follows:

PDP-11 System Building Block (PDP-11 computer)
28K of 900 ns read/write core memory
Dual DECtape transport and Control (TU56)
256K DECdisk and Control (RS11)
Real-Time Clock (Line Frequency) (KW11L)
Hardware Bootstrap Loader
Cabinets and Mounting Hardware

1.1.2 System Software

RSTS-11 system software exists at two levels. The primary level exists in PDP-11 assembly language code and governs the most essential and frequent operations of the system; once the system is generated, this code is "frozen" and not altered thereafter. The secondary level of RSTS-11 software consists of system library programs written in the BASIC-PLUS language and alterable at any time; these programs are called into execution either by the primary system itself or by individual users to perform infrequent functions or functions which can be designed and implemented by the system manager.

1.1.2.1 Primary System Software Elements, Assembly Language Programs

The PDP-11 assembly language elements of the RSTS-11 system are comprised of many distinct modules but can, in general, be classified into generic groups as follows:

(1) permanently core-resident elements:

- (a) interrupt and trap vectors,
- (b) system buffers, large (256 words) and small (16 words),
- (c) system-information data-cells and tables,
- (d) disk and I/O device drivers,
- (e) resident file processor,
- (f) general executive routines,
- (g) on-line Editor (builds text of program lines),
- (h) incremental Compiler (converts BASIC lines into shorter intermediate code stored in the user job area),
- (i) Run-Time System (interpretatively executes intermediate code from user area).

(2) temporarily core-resident elements:

    (j) system initialization code (executed at start-up time and immediately overlaid thereafter by user job areas).

(3) disk-resident elements: (loaded into and executed from an in-core buffer when needed):

    (k) file-processor routines,

    (l) user-core allocation and clean-up routines,

    (m) DECtape processor.

1.1.2.2 Secondary System Software Elements, System Library Programs

The system library programs are actually run in the same manner as a user program; that is, they are executed from the user job area like any other BASIC-PLUS program. System library programs remain, however, under the control of the system manager; and, since they can make extensive use of privileged program status and SYS functions, they exercise wide control over the system. The primary software elements call upon some of these programs to augment their own functions, the system manager will use others to monitor and regulate system usage, and ordinary users can call upon others to perform common utility functions.

1.1.3 System Disk Structures

RSTS-11 V4A contains a logical public disk structure, which consists of a system disk (RF11-type or RK11/RC11-types) and optionally additional public disk packs. All public disks must be on-line whenever the system is running and are accessible to all users; they can be logically distinguished from each other under program control. Elements of the system software are stored on the system disk, including a core image[1] of the core-resident system elements. The system disk is also used as a swapping device for temporary storage of active user jobs until it is their turn to run. Additional space on the system disk as well as all space on other public disks is available for general storage of user programs and data files.

---

[1]Core image is the executable machine code as it would be in core prior to START.

In addition to the public disk structure, RSTS-11 also accommodates private disk structures, i.e. disk packs to which access is granted only to a defined set of users for storage of programs and data files. Private packs can be mounted or dismounted on-line.

1.1.4 General Concepts of System Control

Users active at terminals can be operating either in edit mode or in run mode.

When first logged into the system, the user is in edit mode and returns to edit mode at the completion of any program execution or whenever he types CTRL/C. In edit mode, the system examines each ASCII line entered by the user and determines whether that line is a system command, an immediate-mode statement, or a program statement. System commands are executed immediately. Immediate-mode statements are first translated into an intermediate code, which is placed into the user's job area and then executed immediately by the Run-Time System. Program statements are stored in their ASCII form in a temporary disk file (created by the system with a name of the form TEMPnn.TMP under the user's account) and are also compiled, line by line, into their intermediate code representations and stored in the user job area. The user job area is initialized to a size of 2K words but can grow, as necessary, in 1K increments to a maximum size of 6K (in 24K systems) or 8K (in 28K systems). Program statements are not executed in edit mode, but the program being created can be changed, a copy of the compiled program can be transferred to disk storage (as a .BAC file), or a copy of the ASCII source program can be saved on disk storage (as a .BAS file) or on an external medium.

A user changes from edit mode to run mode when he types the RUN command or the CHAIN immediate mode statement. In run mode, the Run-Time System interpretatively executes the intermediate code stored in the user job area. Following program execution, the user is returned to edit mode. The user can interrupt the Run-Time System by typing CTRL/C; this also returns the user to edit mode.

The system allows user jobs to run (in either edit or run mode) one at a time. A user job will run until it either enters an I/O-wait condition or exhausts the time quantum which the system has assigned to it. At that point the system Scheduler, in round-robin fashion, finds the next job that is ready to run and begins that job. Meanwhile the interrupt-driven I/O handlers are processing the requested data transfers (for all devices except disks and magnetic tape, intermediate system buffers are used for the peripheral transfers) and upon completion of a transfer, mark the job which requested the transfer as again ready to run. In the process of round-robin scanning, the Scheduler will see that the marked job is ready to run again and re-starts it from the point at which execution was last suspended.

The system keeps as many jobs in core as possible, but when more core is required than is available, the system temporarily moves some jobs out of core and stores them in the disk file SWAP.SYS. When it is again their turn to run, these jobs are swapped back into core from the SWAP.SYS file. Thus, jobs waiting for keyboard input and jobs waiting for device I/O completion will likely be stored on the SWAP.SYS file, while jobs currently running and jobs involved in current disk or magnetic tape transfers must necessarily be in core.

As the system processes each job, it maintains accounting information (in core) concerning that job. When the job logs out this information is used to update the accounting information stored on the disk for that account.

1.2 SYSTEM GENERATION

RSTS-11 software is supplied in a form which allows it to be configured easily to any standard RSTS-11 hardware configuration. It also allows choices concerning optional software features which may be included in the system for processing power or may be excluded from the system for core economy. The system manager is advised, therefore, to study Chapter 2 of this manual in order to become thoroughly familiar with the software configuration options that are open to him; he can then select those features and parameters which best suit his applications requirements. Following the directions in Chapter 2, the system manager can create his initial RSTS-11 system and can re-configure and re-create it to meet expanding or altered needs.

1.3 SYSTEM BACK-UP

Once a system has been created, extreme care should be taken to maintain back-up material for it. The supplied RSTS-11 kit itself should be stored in a safe place. Along with the supplied kit, the system CIL DECtape, which is created at system-generation time, should be stored; this DECtape contains the configured system software and makes it possible to re-load that software onto the disk if this should become necessary. The ROLLIN program, which is supplied in the RSTS-11 kit (see Chapter 6), enables the system manager to make image-mode copies of any disk for back-up purposes.

1.4 SYSTEM MANAGEMENT

The RSTS-11 system manager (and all other users holding privileged accounts) is provided with many powers and facilities for controlling the RSTS-11 system.

The system manager assigns the system account numbers and passwords and thereby determines who has privileged access to the system and who has ordinary access.

By retaining a privileged account for himself, the system manager can run the special system library programs which allow him to monitor and reset the system accounting information (with MONEY), to regulate accounts (with REACT), and to govern the use of the system facilities (with UTILTY); and he likewise by-passes any file protection restrictions when he runs any program. The system manager can expand the system library to create his own utility programs to accomplish functions peculiar to the needs of his installation. He has access to a wide selection of SYS functions (see Chapter 4), privileged and otherwise, to facilitate such programming.

CHAPTER 2

RSTS-11 SYSTEM GENERATION

This chapter describes the procedures to generate Version VØ4B-17 RSTS system code from software distributed on 7-track and 9-track magtape, DECtape, and RK disk cartridge (DECpack). Section 2.1 presents an overview of the system generation process and contains comments concerning the media on which DIGITAL distributes the software.

Section 2.2 contains the bootstrap and SYSLOD procedures required for magtape and DECtape software. Section 2.3 describes the bootstrap and copy procedures for disk cartridge software. Procedures to start the system generation batch command file are in Section 2.4. Examples and guidelines to answering configuration questions are in Sections 2.5, 2.6, and 2.7 respectively.

2.1 SYSTEM GENERATION OVERVIEW

The process for generating the RSTS system code consists of the following general steps.

a) If the software is on magnetic tape, bootstrap a tape to load the stand alone program SYSLOD and transfer the system generation monitor to a disk using SYSLOD. If the software is on disk cartridge, bootstrap the system generation disk to load the system generation monitor.

b) Log into the system generation monitor and initiate execution of the batch command file.

c) Answer configuration questions printed during the system generation dialogue.

d) Follow the instructions printed by the system generation monitor and mount and dismount tapes and disks as required.

After the user creates the RSTS system code, he must proceed to Sections 2.1Ø and 2.11 to build the RSTS system disk and system library files.

The user performs the system generation procedure using the DOS/BATCH V09-20 monitor as the system generation monitor. If the DEC-supplied software is on magtape or DECtape, the user must load this monitor onto an RF11 or RK11 disk using the stand alone program SYSLOD. If the DEC-supplied software is on disk cartridge, the user need only bootstrap the cartridge to load the system generation monitor. (In this document, the disk on which the system generation monitor resides is referred to as the system generation disk. Such nomenclature differentiates the disk in question from the disk on which the user eventually builds the RSTS public file structure and which is referred to as the RSTS system disk.)

The procedures for magtape or DECtape distribution software involve transferring files to the system generation disk. After loading the system generation monitor onto a disk, the user must transfer one batch command file from the tape to the disk. The user then initiates execution of the batch commands which transfer all required system generation programs (MACRO, LINK, CILUS, EDIT, PIP, and SYSLOD) from tape to disk without further interaction.

The procedures for disk cartridge distribution software are similar to those for tape except that none of the file transfer operations described are necessary. However, it is advisable that the user copy the disk cartridges containing the system generation and system library programs and use the copies instead of the originals to generate the system. To copy the disk cartridges, the user runs ROLLIN as described in Section 2.3.2 of this guide. To begin system generation, the user bootstraps the copy of the system generation disk cartridge, answers the system generation monitor DIALOGUE questions, and types one command which initiates execution of the batch command file. Commands in the batch file delete any old RSTS system which possibly exists from a prior system generation.

After the system generation monitor transfers all files from tape or deletes files from the disk cartridge, it executes a command in the batch file which runs the system generation program SYSGEN. The program prints approximately 45 hardware and software configuration questions and creates two files based on the answers typed in response to each question. The answers must accurately reflect the hardware configuration on which RSTS will run and the software options desired. During the configuration dialogue with the user, SYSGEN creates the configuration file CF.MAC and a second batch command file SYSGEN.BAT, which are later used to create the RSTS system code tailored to local installation requirements.

After the user answers all the configuration questions, the system generation monitor executes the second batch command file. The monitor conditionally assembles the SY module (system tables), the DK module (disk drivers), and the TT module (terminal service) using the configuration file created during the SYSGEN dialogue.

During execution of the second batch command file, the system generation monitor prints instructions for the user to mount appropriate tapes or disks as each is required. The system generation monitor next links the monitor code, and overlay code (OV) and copies files from tape as needed.

The final step in the system generation process creates a linked core image library (LICIL) of the RSTS system from the load modules created by the linking process. For magtape and DECtape software, the step includes writing the LICIL, system load map, batch and configuration files, and SYSLOD program to a scratch tape. For disk cartridge software, the LICIL is created and remains on the system generation disk.

During the final step for magtape and DECtape software, the monitor prints a message telling the user the exact command to type to write the contiguous core image library, or CIL, onto the RSTS system disk. The batch command file terminates by loading SYSLOD into memory. The user can type the exact command to SYSLOD and write the CIL to the RSTS system disk or, at some later time, can bootstrap the tape to load SYSLOD into memory. After writing the CIL to the RSTS system disk, SYSLOD automatically bootstraps the device to load the RSTS initialization code into memory.

During the final step for disk cartridge software, the system generation monitor prints a message telling the user to mount and write enable the RSTS system disk. When ready, the user types a command to continue executing the batch file which runs CILUS to write the CIL onto the RSTS system disk without further user action. If the RSTS system disk is an RF11 disk, the batch command file terminates by bootstrapping that device. If the RSTS system disk is an

RK disk, the system generation monitor prints a message telling the user to move the disk to RK unit Ø and to bootstrap it by the hardware loader to load the RSTS initialization code.

The RSTS Core Image Library contains the system initialization code, the monitor, the BASIC-PLUS Run Time System, overlay code, error messages, and, optionally, the stand alone programs ROLLIN and DSKINT. When the RSTS system disk is bootstrapped, the initialization code is loaded into memory. The user must install necessary patches (PATCH option), create the system files (REFRESH option), and begin time sharing (START option).

2.2 MAGTAPE AND DECTAPE PROCEDURES

Magtape and DECtape procedures differ in the bootstrap procedures required and in the device designators used in several keyboard commands.

2.2.1 Magtape Bootstrap Procedure

Bootstrapping a magtape involves using the central processor unit (CPU) console switches and the MR11-DB Bulk Storage Bootstrap Loader. If the MR11-DB is not on the system, the user must deposit a loading routine manually from the console switch register. This action is necessary since the BM792-YB hardware loader cannot bootstrap a magtape.

The CPU console switches are described in Chapter 8 of the PDP-11/45 Processor Handbook and the PDP-11/4Ø Processor Handbook and in Chapter 7 (Part I) of the PDP-11/2Ø Processor Handbook. Use of the magtape transport is described in Section 5.6 of the RSTS-11 System User's Guide.

To bootstrap the magtape, perform the following steps.

Move the CPU Console ENABLE/HALT Switch to its HALT position.

Mount the magnetic tape reel labelled DEC-11-ORTBB-A-MC7 or -MC9 (SYSTEM GENERATION AND LIBRARY TAPE) on unit Ø with the write enable ring removed.

Ensure that the tape is at its load point. (The LD PT indicator comes on.) The computer does not bootstrap the device unless the tape is at its load point.

Set the ON-LINE/OFF LINE switch on the tape unit to ON-LINE and ensure that the RDY indicator is lit.

Ensure that the console terminal is on line.

If the MR11-DB loader is on the system, proceed to Section 2.2.1.1. If the system configuration does not include the MR11-DB loader, go to Section 2.2.1.2.

2.2.1.1 MR11-DB Procedure - If the MR11-DB is on the system, perform the following steps.

Set the CPU console Switch Register to 773136.

Depress the CPU console LOAD ADRS switch.

Move the CPU console ENABLE/HALT switch to its ENABLE position.

Depress the CPU console START switch.

Proceed to Section 2.2.3 to transfer the system generation monitor from tape to disk.

2.2.1.2 Systems Without the MR11-DB Loader - If the system configuration does not include the MR11-DB hardware loader, the user must manually enter the following load routine using the CPU Console Switch Register and DEP Switch. To load the routine, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position.

Set the Switch Register to Ø1ØØØØ.

Depress the CPU LOAD ADRS switch.

Load the following contents into memory using the Switch Register and DEP switch.

Address	Contents
Ø1ØØØØ	Ø127ØØ
Ø1ØØØ2	172524
Ø1ØØØ4	ØØ531Ø
Ø1ØØØ6	Ø1274Ø
Ø1ØØ1Ø	Ø6ØØ11
Ø1ØØ12	1Ø571Ø
Ø1ØØ14	1ØØ376
Ø1ØØ16	ØØ571Ø
Ø1ØØ2Ø	1ØØ767
Ø1ØØ22	Ø1271Ø
Ø1ØØ24	Ø6ØØØ3
Ø1ØØ26	1Ø571Ø
Ø1ØØ3Ø	1ØØ376
Ø1ØØ32	ØØ571Ø
Ø1ØØ34	1ØØ777
Ø1ØØ36	ØØ5ØØ7

Set the Console Switch Register to Ø1ØØØØ.

Depress the CPU Console LOAD ADRS switch.

Depress the CPU Console START switch.

Proceed to Section 2.2.3 to transfer the system generation monitor from tape to the disk.

2.2.2 DECtape Procedures

Bootstrapping a DECtape involves using the central processing unit (CPU) console switches and either the MR11-DB or BM792-YB hardware bootstrap loader. The CPU switches are described in Chapter 8 of either the PDP-11/45 Processor Handbook or the PDP-11/40 Processor Handbook and in Chapter 7 (Part I) of the PDP-11/20 Processor Handbook. Use of the DECtape transport is described in Section 5.5 of the RSTS-11 System User's Guide.

To bootstrap the DECtape, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position.

Mount the DECtape reel labelled DEC-11-ORDBA-A-UB (DOS MONITOR) on unit Ø.

On DECtape unit Ø, set the REMOTE/OFF/LOCAL switch to REMOTE and the WRITE ENABLE/WRITE LOCK switch to WRITE LOCK.

Ensure that the console terminal is on line.

Proceed to Section 2.2.2.1 if the MR11-DB is on the system. Otherwise, follow the procedures in Section 2.2.2.2.

2.2.2.1 MR11-DB Procedures - If the MR11-DB Bootstrap Loader is on the system,

Move the CPU Console ENABLE/HALT switch to its ENABLE position.

Set the CPU Console Switch Register to 77312Ø.

Depress the CPU Console LOAD ADRS switch.

Depress the CPU Console START switch.

Proceed to Section 2.2.3 to transfer the system generation monitor from tape to the disk.

2.2.2.2 BM792-YB Procedures - If the BM792-YB Bootstrap Loader is on the system,

Set the CPU Console Switch Register to 7731ØØ.

Depress the CPU Console LOAD ADRS switch.

Move the CPU Console ENABLE/HALT switch to its ENABLE position.

Set the CPU Console Switch Register to 777344.

Depress the CPU Console START switch.

Proceed to Section 2.2.3 to transfer the system generation monitor from tape to the disk.

2.2.3 Loading the System Generation Monitor From Tape

After the user follows the magtape or DECtape bootstrap procedures described in Sections 2.2.1 and 2.2.2, the computer reads unit Ø and loads SYSLOD into memory. SYSLOD prints its identification line followed by the first in a series of queries as follows.

```
SYSLOD VØ7-Ø2A
CONSOLE FILL COUNT=
```

If SYSLOD does not print its identification and the processor halts, a parity error possibly was detected in reading the tape. Retry the entire procedure, including rewinding the tape to its load point. (When using magtape on systems without the MR11-DB hardware loader, verify the accuracy of the loading routine by use of the CPU Console EXAM switch before retrying the procedure. Use the DEP switch and the CPU Console Switch Register to correct any errors. If SYSLOD fails to identify itself, it will be necessary to obtain a new magtape reel.)

2.2.3.1 Answering the SYSLOD Questions - When SYSLOD runs, perform the following steps.

If the system generation disk is either an RKØ3 or RKØ5 cartridge, mount it on drive unit Ø.

Ensure that the system generation disk is on line, write enabled, and ready before proceeding. (After the DIALOGUE query is answered, SYSLOD does not recognize any devices previously not ready.)

Type the RETURN key in response to the CONSOLE FILL COUNT question and proceed as follows:

```
SYSLOD VØ7-2A
CONSOLE FILL COUNT=         (Type RETURN key.)
DATE:  14-JUL-74            (Type in dd-mmm-yy format.)
DIALOGUE?                   (Type RETURN key.)
#
```

Type the current date in the standard dd-mmm-yy format and type the RETURN key in response to the DIALOGUE query.

SYSLOD indicates that it is ready to accept a command string by printing the # character. A single command string is necessary to format the disk, to check for bad blocks, and to transfer the system generation monitor to the disk. Use the following format for the SYSLOD command string.[1]

```
#xx:MONLIB.CIL/FO/CO:2/HO/BO<yy:MONLIB.LCL
```

where:

xx is DK for an RKØ5 or RKØ3 disk cartridge
DF for an RF-type disk.

NOTE

The /FO switch is not necessary for RF-type disks.

yy is DTØ if DECtape software is used, or
MTØ if either 7- or 9-track magtape software is used.

The /FO switch in the SYSLOD command causes a removable disk to be formatted. The /CO:2 switch writes two patterns on each block of the disk to ensure that no bad blocks are used. The /HO switch causes SYSLOD to place a pointer to the CIL in the bootstrap record. The /BO switch causes SYSLOD to bootstrap the device upon completing the transfer.

The entire process takes between 5 and 15 minutes depending upon the size and type of disk. If SYSLOD prints any error messages, con-

[1]On an LA3Ø(S) DECwriter and a VTØ5 alphanumeric display terminal, it may be necessary to use the SHIFT key while typing alphabetic characters in order to insure that upper case characters are transmitted.

sult Appendix C for the meaning and possible steps for recovery. Upon completing the transfer, SYSLOD prints the following messages.

```
SYSLOD COMPLETE

ANSWER WITH CARRET OR 'Y' CARRET: -IS YOUR LINE FREQUENCY 5Ø HERTZ?
DO YOU WANT TO DISABLE DIALOGUE FOREVER?  NO
DOS/BATCH VØ9-2Ø
DATE:
```

Type the RETURN key (CARRET for carriage return) if the line frequency of the power used to run the PDP-11 is 6Ø Hertz. Otherwise, type Y and the RETURN key to indicate 5Ø Hertz.

SYSLOD then prints a question asking if the user wants to disable the dialogue forever. The system generation monitor begins with a dialogue similar to that used by SYSLOD. It is possible to disable this dialogue at this time by typing Y followed by the RETURN key. For RSTS system generation purposes, type NO so that the dialogue is not disabled. Proceed to Section 2.4.1 for instructions on answering the system generation monitor dialogue.

2.3 DISK CARTRIDGE PROCEDURES

Disk cartridge procedures involve bootstrapping the device and copying the original distribution cartridges using the stand alone program ROLLIN.

To prevent possible destruction of the system generation and system library disk cartridges, it is advisable to copy the cartridges and use the copies for system generations. The cartridges are created on properly aligned drives. Since drive alignment drifts slightly in shipping and with age, problems possibly occur at a user site. If the user cannot copy the cartridges, DIGITAL Field Service must check the drive alignment before system generation can continue. The stand alone program ROLLIN is included on the system generation disk cartridge to facilitate the copy operation.

2.3.1 Disk Cartridge Bootstrap

Bootstrapping a disk cartridge involves using the central processor unit (CPU) console switches and either the MR11-DB or BM792-YB hardware loader. The CPU console switches are described in Chapter 8 of the PDP-11/45 Processor Handbook and the PDP-11/4Ø Processor Handbook and in Chapter 7 (Part I) of the PDP-11/20 Processor Handbook.

To bootstrap the cartridge, perform the following steps.

Physically mount the cartridge labelled DEC-11-ORDPB-A-HC (SYSTEM GENERATION) in the RKØ3 or RKØ5 unit Ø.

Ensure that the RDY light is on.

If the cartridge is not yet copied, ensure that the WR PROT light is on. (This condition write protects the disk.) If the copying is complete, ensure that the WR PROT light is off. On an RKØ5 drive, depressing the WR PROT switch alternately turns the WR PROT light on and off.

Ensure that the console terminal is on line.

Move the CPU Console HALT/ENABLE switch to its HALT position and back to its ENABLE position.

If the MR11-DB Bulk storage bootstrap loader is on the system, go to Section 2.3.1.1. Otherwise, go to Section 2.3.1.2.

2.3.1.1 MR11-DB Procedure - If the MR11-DB is on the system perform the following steps.

Set the CPU Switch Register to 77311Ø.

Depress the CPU LOAD ADRS switch.

Depress the CPU START switch.

Proceed to Section 2.3.1.3.

2.3.1.2 BM792-YB Procedure - If the BM792-YB Hardware Loader is on the system, perform the following steps.

Set the CPU Switch Register to 7731ØØ.

Depress the CPU LOAD ADRS switch.

Set the CPU Switch Register to 7774Ø6.

Depress the CPU START switch.

Proceed to Section 2.3.1.3.

2.3.1.3 Monitor Identification - The system reads the disk and loads the system generation monitor which prints the following lines.

```
DOS/BATCH VØ9-2Ø
DATE:
```

If the monitor fails to identify itself, retry the entire procedure, and carefully check the switch register data. After the monitor prints its identifying lines, proceed to Section 2.3.2 to copy the system generation and system library cartridges or proceed to Section 2.4 to start system generation.

2.3.2 Copying the Disk Cartridges Using ROLLIN

To copy the disk cartridges, perform the following steps.

Mount a new disk cartridge on drive unit 1.

Ensure that the RDY light comes on and that the WR PROT light for unit 1 is off.

Ensure that the WR PROT light for unit Ø is on. The original disk must be write protected to prevent inadvertent destruction.

Continue the dialogue in the following manner.

```
DOS/BATCH VØ9-2Ø
DATE:  11-JUL-74          (Type in dd-mmm-yy format.)
TIME:  Ø6:51              (Type in hh:mm format.)
DIALOGUE?                 (Type the RETURN key)
```

When the monitor prints the $ character, type the LO 1,1 command. The monitor prints the current date and time followed by the $ character.

```
$LO 1,1                   (Terminate with RETURN key.)
DATE:  11-JUL-74
TIME:  Ø6:52
$
```

The user must run the CILUS program to load ROLLIN. Type the RUN CILUS command as shown.

```
$RUN CILUS                (Terminate with RETURN key.)
CILUS VØ8-Ø6A
#
```

CILUS runs and prints its header line followed by the # character. Type the command shown to run ROLLIN and subsequently to copy unit Ø to unit 1.

```
#ROLLIN.CIL/BO          (Terminate with RETURN key.)

ROLLIN VØ7
#DK1:/FO<DKØ:/VE        (Terminate with RETURN key.)
```

ROLLIN prints messages signalling the start and end of the format pass and the start of the verification pass. If no errors are encountered, ROLLIN prints the # character again as shown below.

```
STARTING RK FORMAT PASS
END RK FORMAT PASS
STARTING VERIFICATION PASS
#
```

If any errors are encountered, ROLLIN prints appropriate messages and the # character. The user must contact a Field Service representative to align the drive. If ROLLIN does not print any error messages, the user can continue according to the following steps.

Move the LOAD/RUN switch to its LOAD position on both units Ø and 1.

When the LOAD light comes on, remove the cartridges from their respective drives.

Label the copied cartridge in such manner as SYSTEM GENERATION COPY. Store the original in a safe place.

Mount the disk cartridge labelled DEC-11-ORPTB-A-HA (SYSTEM LIBRARY) in unit Ø. Ensure that the WR PROT light is on.

Mount a second new cartridge in unit 1. Ensure that the RDY light comes on and that the WR PROT light for unit 1 is off. Ensure that the WR PROT light for unit Ø is on. The original disk on unit Ø must be write protected.

Since ROLLIN is still waiting, type the following command in response to the # character.

```
#DK1:/FO<DKØ:/VE
STARTING RK FORMAT PASS
END RK FORMAT PASS
STARTING RK VERIFICATION PASS
#
```

If any errors are encountered, ROLLIN prints appropriate messages and the # character. The user must contact a Field Service representative to align the drive. If ROLLIN does not print any error messages, the user can continue according to the following steps.

Move the LOAD/RUN switch to its LOAD position on both units Ø and 1.

When the LOAD light comes on, remove the cartridges from their respective drives.

Label the copied cartridge in such manner as SYSTEM LIBRARY COPY. Store the original system library cartridge with the original system generation cartridge.

Mount the copied system generation disk in unit Ø and move the LOAD/RUN switch to its RUN position. Ensure that the RDY light comes on and that the WR PROT light is off. (The disk must be write enabled.)

Bootstrap unit Ø by typing the following command to ROLLIN.

```
#/BO:DK
DOS/BATCH VØ9-2Ø
DATE:
```

Proceed to Section 2.4.1 to start the system generation monitor.

2.4 STARTING SYSTEM GENERATION

Once the system generation disk is bootstrapped, the system generation monitor runs. The user must perform the monitor dialogue and the login procedure; and initiate the batch command file. If magtape or DECtape distribution software is used, he must additionally transfer the batch command file from tape to the disk.

2.4.1 Monitor Dialogue

After the disk is bootstrapped, the system generation monitor prints its identification line followed by the first of several prompting lines. Type the current date in response to DATE: and the current time of day in response to TIME: as shown below. Terminate each response with the RETURN key.

```
DOS/BATCH VØ9-2Ø

DATE:  12-JUL-74          (Use dd-mmm-yy format.)
TIME:  1Ø:12              (Use hh:mm format.)
DIALOGUE?
```

The user can omit the monitor dialogue if the line printer used for system generation is an LP11 with 8Ø columns and if the console fill count required is Ø.

To omit the dialogue, type the RETURN key in response to the DIALOGUE query. The monitor prints the $ character after which the user can continue at Section 2.4.2 to perform the login procedure.

To include the dialogue type YES followed by the RETURN key in response to the DIALOGUE query and proceed as shown.

```
DIALOGUE?  YES
DO YOU WANT TO SET CONSOLE FILL COUNT?  YES
FILL COUNT=
```

Type the console fill count in response to the FILL COUNT= query according to the following values for the type of console terminal on the computer.

Console Fill Values	Console Terminal Types
Ø	ASR-33 and ASR-35 Teletype; LA3ØS and LA3ØP DECwriter (110 and 15Ø baud)
	VTØ5 display (11Ø, 15Ø, and 3ØØ baud)
1	ASR-37 Teletype
	VTØ5B display at 6ØØ baud
4	LA3ØP DECwriter at 3ØØ baud
12	LA3ØS DECwriter at 3ØØ baud

For example, type 12 for an LA3ØS DECwriter at 3ØØ baud. Type the RETURN key in response to the remaining questions as shown below unless the line printer is not an LP11 with 8Ø columns.

```
FILL COUNT=12
ARE ANY DEVICES DOWN?
DO YOU WANT TO CHANGE LINE PRINTER?
```

The question concerning the card reader appears only if the system has such a device. The monitor prints the $ character in response to which the user must perform the login procedure described in Section 2.4.2.

To change the line printer default, type YES in response to the line printer question and proceed as shown below.

```
DO YOU WANT TO CHANGE LINE PRINTER?  YES
LS11?  NO
HOW MANY COLUMNS?  132
LOWER CASE?  NO
OVERPRINT?  NO
```

If unit Ø is an LS11 dot-matrix line printer, type YES in response to the LS11 query. Otherwise, type NO and the RETURN key to indicate an LP11 line printer. Type the number of columns in response to the COLUMNS query. Type NO and the RETURN key in response to the LOWER CASE, OVERPRINT and remaining queries and proceed as described below when the $ character appears.

2.4.2 Performing the Login Procedure

To log into the system generation monitor, type the LO 1,1 command in response to the $ character as shown below.

```
$LO 1,1
DATE:  21-JUL-74
TIME:  1Ø:56
$
```

If the login procedure is done properly, the monitor prints the current date and time followed by the $ character. Otherwise, the monitor prints an appropriate error message followed by the $ character. The user must try again.

At this point, procedures differ slightly for tape and disk software. Continue with Section 2.4.3 if the software is on either magtape or DECtape. Proceed to Section 2.4.4 if the software is on disk cartridge.

2.4.3 Transferring the Batch Command File from Tape

The user must execute the PIP program from tape and type a command to transfer the batch command file to the system generation disk. Use the following format for the command to execute PIP.

```
$RUN xx:PIP
```

where:

xx is MT for either 7-track or 9-track magtape or
DT for DECtape.

When PIP prints the identification line and the # character as follows:

```
PIP V1Ø-Ø2
#
```

use the following format to transfer the file.

```
# SY:<xx:SYSGEN
#
```

where:

xx is MT for either 7-track or 9-track magtape or
DT for DECtape.

PIP signals completion by printing the # character again. Type the CTRL/C combination to terminate PIP and the KI command in response to the dot character printed by the monitor. For example,

```
#↑C
.KI
$
```

When the monitor prints the $ character, proceed to Section 2.4.4 to execute the batch command file.

2.4.4 Initiating the Batch Command File

To initiate execution of the first batch command file, type the following command in response to the $ character.

```
$BATCH SYSGEN
```

The monitor executes the commands in the batch command file SYSGEN. When the following message and question is printed, SYSGEN has entered the configuration dialogue.

```
RSTS-11 V4B SYSTEM GENERATION

DO YOU WANT QUESTIONS IN THE SHORT FORM(S)
OR LONG (ANYTHING ELSE):
```

Section 2.5 contains an explanation of the various forms of the questions.

2.5 CONFIGURATION QUESTIONS

After the user initiates the batch command file, the system generation program SYSGEN runs and enters a dialogue with the user. The dialogue is a series of approximately 45 hardware and software configuration questions. The questions come in both a long and a short form.

Long form questions contain explanatory information and are useful to users who are unfamiliar with the system. For a sample printout of the long form questions, see Section 2.6.1.

If the user is familiar with the dialogue questions, and wishes to save time, he can select the short form of the questions. A sample printout of the short form questions appears in Section 2.6.2. A tabulation of the short form questions and possible answers with comments is given in Appendix A.

During the dialogue, SYSGEN checks the answers which the user gives. If an answer is incorrect, SYSGEN reprints the query or series of queries regarding that subject. Implications of the configuration questions are given in Section 2.7.

After the user answers the configuration questions, the monitor begins executing the second batch command file. For information on this part of the procedure, consult Section 2.8.

2.6 SYSTEM GENERATION EXAMPLES

This section contains two samples of system console terminal printout produced during the generation of RSTS. The right hand margin of each sample has references to other sections which contain relevant descriptions. The samples show the system generation to the point where the software automatically bootstraps the resultant RSTS system disk and loads the initialization code into memory for the first time.

The first sample shows the output produced by using magtape software with an RKØ5 disk cartridge as the system generation disk and an RKØ5 disk cartridge as the resultant RSTS system disk. The sample shows all configuration questions in long form.

The second sample shows a system generation using disk cartridge (DECpack) software. The system generation disk and the resultant RSTS system disk are RKØ5 disk cartridges. The sample shows short form configuration questions.

2.6.1 Magtape Distribution Software Using Long Form Questions

```
RSTS-11 V4B SYSTEM GENERATION

DO YOU WANT QUESTIONS IN THE SHORT FORM (S)                  (2.4.4)
OR LONG (ANYTHING ELSE): LONG

YOU WILL BE ASKED A SERIES OF QUESTIONS                      (2.5)
ABOUT YOUR PARTICULAR SYSTEM CONFIGURATION.
EACH QUESTION WILL GIVE YOU THE PROPER
RESPONSES. IF YOUR RESPONSE IS INVALID, THEN
THE QUESTION WILL BE REPEATED AND YOU MAY TRY AGAIN
FOR A VALID RESPONSE. YOU SHOULD CONTINUE TO USE
THE LONG FORM OF THE QUESTIONS UNTIL YOU ARE
VERY FAMILIAR WITH THE QUESTIONS AND
THEIR ANSWERS. WHEN ALL QUESTIONS ARE DONE,
THE CONFIGURATION FILE 'CF.MAC' WILL BE WRITTEN OUT
FOR USAGE IN ASSEMBLING THE RSTS FILES.

A BATCH CONTROL FILE 'SYSGEN.BAT' WILL BE
WRITTEN OUT THAT WILL COMPLETE THE GENERATION
PROCEDURE.

ENTER YOUR SYSTEM INSTALLATION NAME. IT
MAY BE FROM 1 TO 14 CHARACTERS LONG.
I WILL DRAW A LINE TO SHOW YOU HOW LONG A 14
CHARACTER NAME WOULD BE: --------------
PLEASE ENTER NAME HERE:  TEST 104 MT

ENTER THE AC POWER FREQUENCY IN HERTZ. THE                   (2.7.1)
ONLY RESPONSES ARE '60' AND '50'.
ENTER YOUR FREQUENCY: 60

IS YOUR CLOCK A KW11L (STANDARD LINE                         (2.7.2)
FREQUENCY CLOCK) OR A KW11P (PROGRAMMABLE
CLOCK). ANSWER 'L' FOR A KW11L OR 'P' FOR A
KW11P. YOURS IS: L

IF YOU WISH AUTOMATIC POWER FAIL RECOVERY,                   (2.7.3)
THEN ANSWER THIS QUESTION WITH A 'Y'; ELSE
ANSWER WITH A 'N'.
HOW ABOUT YOU: Y

ENTER THE HIGHEST JOB NUMBER YOU WANT FOR YOUR               (2.7.4)
SYSTEM. IT MUST BE BETWEEN 1 AND 17.
YOUR HIGHEST JOB NUMBER IS: 10

ENTER NUMBER OF KL11 PLUS DL11A PLUS DL11B PLUS LC11         (2.7.5)
TYPE INTERFACES ON THE SYSTEM. NOTE THAT THIS VALUE
DOES NOT INCLUDE THE CONSOLE'S INTERFACE. THE VALUE,
THEREFORE, HAS A RANGE OF 0 TO 16.
YOU HAVE HOW MANY INTERFACES: 1

ENTER NUMBER OF DC11 TYPE INTERFACES ON THE SYSTEM.
THE VALUE IS BETWEEN 0 AND 16.
YOU HAVE HOW MANY DC11 TYPE INTERFACES: 0
```

```
ENTER NUMBER OF DL11E TYPE INTERFACES ON THE SYSTEM.
THE VALUE IS BETWEEN 0 AND 16.
YOU HAVE HOW MANY DL11E TYPE INTERFACES: 1

IF YOU DESIRE THE GENERALIZED TERMINAL FILL FEATURE          (B.3)
OF THE TERMINAL SERVICE, THEN ANSWER THIS QUESTION
WITH 'Y'; ELSE ANSWER WITH 'N' FOR NO GENERALIZED
FILL. YOU WANT: Y

IF YOU HAVE ANY SERIAL LA30 TYPE TERMINALS, THEN
ANSWER THIS QUESTION WITH 'Y'; ELSE IF YOU DON'T
HAVE ANY SERIAL LA30'S, THEN ANSWER WITH 'N'.
ANY SERIAL LA30'S: Y

IF YOU HAVE UPPER/LOWER CASE TERMINALS,                      (B.2)
YOU MAY WANT RSTS TO BE ABLE TO SELECTIVELY
TRANSLATE LOWER CASE CHARACTERS TO UPPER
CASE TO AVOID POSSIBLE PROBLEMS WITH STRING
COMPARISONS.  IF YOU WANT THIS FEATURE, ANSWER
'Y', ELSE ANSWER 'N'   YOUR ANSWER: Y

SOME NEWER TERMINALS TREAT THE ASCII CHARACTERS              (B.1)
175 & 176 AS REAL CHARACTERS, WHILE OLDER TERMINALS
TREAT THESE AS ESCAPE CHARACTERS.  YOUR RSTS SYSTEM
CAN BE GENERATED TO SELECTIVELY DISABLE RECOGNITION
OF THESE CHARACTERS AS ESCAPE.  IF YOU WANT THIS
FEATURE, ANSWER 'Y', ELSE ANSWER 'N'.  YOUR
ANSWER IS: Y

ANSWER THIS QUESTION WITH 'Y' IF YOU HAVE REMOTE             (B.4)
TERMINALS SUCH AS AN ASR33 THAT HAVE THE LOCAL
READER CONTROL FEATURE CALLED XON(+XOFF). ANSWER
WITH 'N' IF YOU DO NOT WANT THE XON FEATURE IN YOUR
SYSTEM. YOU WANT: Y

IF YOU WANT THE TERMINALS THAT CALL IN TO YOUR               (B.6)
DATASET TYPE INTERFACES TO GET AN AUTOMATIC ANSWER
MESSAGE, THEN ANSWER THIS QUESTION WITH 'Y'; ELSE
ANSWER 'N' AND THEY WILL GET NO MESSAGE.
DO YOU WANT MESSAGES: Y

IF YOU HAVE ANY SCOPE TYPE TERMINALS, THEN ANSWER
QUESTION WITH 'Y'; ELSE ANSWER 'N' AND YOU WILL NOT
HAVE THE TERMINAL SERVICE SUPPORT FOR SCOPE TYPE
TERMINALS. SCOPE SUPPORT: Y

THIS MULTIPLE CHOICE QUESTION IS ABOUT PARITY
GENERATION FOR TERMINALS. YOUR ANSWER AFFECTS ALL            (B.5)
TERMINALS:
'1'      NO PARITY GENERATION (8 BITS SENDABLE)
'2'      EVEN PARITY GENERATED
'3'      ODD PARITY GENERATED
YOUR CHOICE: 1

IF YOU DON'T HAVE THE RF11 256K FIXED HEAD DISK,             (2.7.6)
THEN ANSWER WITH A '0'. IF YOU DO HAVE THE RF11 DISK,
ANSWER WITH THE NUMBER OF RS11 256K PLATTERS THAT ARE
ON THE RF11 CONTROL(1 TO 8).
YOUR ANSWER: 2
```

IF YOU HAVE THE RK11 CARTRIDGE DISK CONTROL ON
YOUR SYSTEM, THEN ANSWER WITH THE NUMBER OF RK03
PLUS RK05 DRIVES THAT YOU HAVE(1 TO 8). IF YOU DON'T
HAVE THE RK11 ANSWER WITH A 0.
YOUR ANSWER IS: 2

SINCE YOU HAVE BOTH AN RF11 AND AN RK11 DISK, EITHER MAY BE USED AS THE RSTS-11 SYSTEM DISK. IF YOU ANSWER THIS QUESTION 'Y', YOUR SYSTEM DISK WILL BE RK UNIT 0, AND THE RF DISK WILL BE USED ONLY FOR SWAPPING. IF YOU ANSWER 'N', THE RF11 WILL BE USED FOR BOTH SYSTEM AND SWAPPING. (2.7.7)
YOUR ANSWER: Y

IF YOU HAVE A HIGH SPEED PAPER TAPE READER, THEN ANSWER 'Y'; ELSE ANSWER 'N'. (2.7.8)
YOUR ANSWER IS: Y

IF YOU HAVE A HIGH SPEED PAPER TAPE PUNCH, THEN ANSWER 'Y'; ELSE ANSWER 'N'.
YOUR ANSWER IS: Y

ENTER NUMBER OF TU56 DUAL DECTAPE DRIVES ON THE SYSTEM.. THIS WOULD BE A NUMBER FROM 1 TO 4 CORRESPONDING TO 2 TO 8 DECTAPE DRIVES. IF YOU HAPPEN TO HAVE NO DECTAPE AT ALL THEN ANSWER 0.
YOUR NUMBER OF TU56 DUAL DRIVES IS: 1

ENTER THE NUMBER OF BIG (256 WORD) BUFFERS FOR YOUR SYSTEM. THE NUMBER IS FROM 1 TO THE NUMBER OF DECTAPE DRIVES (2.7.9)
NUMBER OF BIG BUFFERS: 1

IF YOU HAVE A CARD READER (BE IT THE CR11 PUNCHED CARD READER OR THE CM11 MARK SENSE CARD READER) THEN ANSWER 'Y'; ANSWER 'N' FOR NO CARD READER. YOUR ANSWER: N (2.7.10)

IF YOU HAVE THE LP11 LINE PRINTER, THEN ANSWER WITH A 'Y'; ELSE ANSWER 'N' FOR NO LINE PRINTER.
YOUR ANSWER: Y

IS YOUR LINE-PRINTER A CENTRONICS (LS11) LINE PRINTER: Y

IF YOU HAVE THE TM11 MAGTAPE CONTROL, THEN ANSWER WITH THE NUMBER OF TU10 DRIVES (7 TRACK PLUS 9 TRACK). IF YOU HAVE NO MAGTAPE, THEN ANSWER WITH A 0. THE RANGE OF NUMBER OF DRIVES IS FROM 1 TO 8. NUMBER OF TU10 DRIVES: 2

ENTER THE NUMBER OF SMALL (16 WORD) BUFFERS YOU WISH TO HAVE FOR YOUR SYSTEM. 64 SMALL BUFFERS SEEMS TO BE A GOODLY NUMBER. THE RANGE IS FROM 10 TO 999 SMALL BUFFERS. THE NUMBER OF SMALL BUFFERS FOR OPTIMUM SYSTEM OPERATION IS A FUNCTION OF THE NUMBER OF JOBS POSSIBLE AND HOW MANY FILES THOSE JOBS WILL HAVE OPEN AT ONCE. NUMBER OF SMALL BUFFERS: 75 (2.7.11)

```
SELECT YOUR MATH PACKAGE FROM THE LIST.                          (2.7.12)

2    STANDARD 2-WORD MATH
2E   2-WORD EAE MATH
2X   STANDARD 2-WORD WITH NO FUNCTIONS
2EX  2-WORD EAE WITH NO FUNCTIONS
4    STANDARD 4-WORD MATH
4E   4-WORD EAE MATH
4X   STANDARD 4-WORD WITH NO FUNCTIONS
4EX  4-WORD EAE WITH NO FUNCTIONS
ENTER MATH PACKAGE DESIRED: 4

YOUR RSTS-11 SYSTEM CAN PRINT THE TIME
OF DAY IN 24-HOUR MILITARY FORMAT OR
IN STANDARD AM/PM FORMAT.  IF YOU
ANSWER 'Y',  24-HOUR TIME WILL BE
USED. IF YOU ANSWER 'N', AM/PM WILL
BE USED.  THE DEFAULT IS AM/PM.  DO YOU
WANT 24-HOUR TIME: N

THE FOLLOWING QUESTIONS ARE ABOUT OPTIONAL SPECIAL
FEATURES. IF YOU WANT THE FEATURE DESCRIBED, THEN
ANSWER WITH A 'Y'; ELSE ANSWER WITH 'N' TO OMIT
THE FEATURE.

ALL RSTS V4B SYSTEMS GET THE AUTOMATIC SYSTEM RELOAD
AND RE-START FOLLOWING A SYSTEM ERROR, BUT THERE ARE
SPECIAL ROUTINES THAT ATTEMPT TO RECOVER FROM MOST OF
THESE ERRORS WITHOUT AFFECTING MORE THAN THE USER THAT
CAUSED THE ERROR. DO YOU WANT THESE ROUTINES: Y

THE RECORD I/O FEATURE ENABLES THE VERBS:                        (2.7.13)
 LSET,RSET,FIELD,GET,PUT,CVT,XLATE.
DO YOU WANT THE RECORD I/O FEATURE: Y

DO YOU WANT THE FILE 'UPDATE' MODE FEATURE: Y                    (2.7.14)

DO YOU WANT 'PRINT-USING' INCLUDED: Y                            (2.7.15)

DO YOU WANT THE 'MAT' COMMANDS: Y                                (2.7.16)

IF YOU WISH LINE PRINTER LISTINGS OF THE RSTS                    (2.7.17)
ASSEMBLIES THAT ARE TO FOLLOW, THEN ANSWER WITH
A 'Y'; ELSE ANSWER WITH 'N' FOR NO LISTINGS.
YOUR ANSWER: N

PLEASE INDICATE YOUR RSTS-11
DISTRIBUTION MEDIA FROM THE LIST
BELOW:
        DT      DEC-TAPE
        MT      MAG-TAPE
        DK      DEC-PACK
YOUR MEDIA IS: MT

THERE ARE TWO OPTIONAL MODULES THAT YOU CAN INCLUDE               (2.7.18)
AND SAVE ON YOUR RSTS SYSTEM DISK FOR LATER CALLING.
THEY ARE:
DSKINT  RSTS V4B DISK INITIALIZATION PROGRAM
        INIT IS USED TO CREATE RSTS V4B NON-SYSTEM
        DISKS.
ROLLIN  ROLLIN-ROLLOUT DISK+DECTAPE+MAGTAPE COPY
        ROLLIN IS USED TO DUMP OR LOAD DISKS TO OR
        FROM EITHER DECTAPE OR MAGTAPE.
```

```
ANSWER THE QUESTION ABOUT THE MODULE'S NAME WITH
EITHER 'Y' TO INCLUDE THE MODULE OR 'N' TO OMIT
THE MODULE.

INCLUDE MODULE 'DSKINT': Y

INCLUDE MODULE 'ROLLIN': Y
```

2.6.2 Disk Cartridge Distribution Software Using Short Form Questions

```
RSTS-11 V4B SYSTEM GENERATION

DO YOU WANT QUESTIONS IN THE SHORT FORM <S>             (2.4.4)
OR LONG <ANYTHING ELSE>: S                              (2.5)

NAME OF INSTALLATION: TEST 104 RK

AC FREQUENCY: 60                                        (2.7.1)

CLOCK: L                                                (2.7.2)

POWER FAIL: Y                                           (2.7.3)

MAX JOB NUMBER: 10                                      (2.7.4)

KL11+DL11A+DL11B+LC11'S: 2                              (2.7.5)

DC11'S: 1

DL11E'S: 1

FILL: Y                                                 (B.3)

SERIAL LA30'S: Y

LOWER CASE: Y                                           (B.2)

ESCAPE: Y                                               (B.1)

XON: Y                                                  (B.4)

AUTO ANSWER: Y                                          (B.6)

SCOPES: Y

PARITY: 1                                               (B.5)

RF11: 2

RK11: 2                                                 (2.7.6)

SWAP ONLY RF: Y                                         (2.7.7)

PT READER: Y                                            (2.7.8)

PT PUNCH: Y

TC11: 1
```

```
BIG BUFFERS: 1                                  (2.7.9)
CR11: N
LP11: Y
LS11 TYPE: N
LP COLUMNS: 132
LP LC: N
LP FORMAT: Y
TM11: 2
SMALL BUFFERS: 75                               (2.7.11)
MATH PACKAGE: 2                                 (2.7.12)
24-HOUR TIME: N
NO FAIL: Y
RECORD I/O: Y                                   (2.7.13)
UPDATE: Y                                       (2.7.14)
PRINT-USING. Y                                  (2.7.15)
MATRIX: Y                                       (2.7.16)
LP LISTINGS: N                                  (2.7.17)
MEDIA: DK
DSKINT: Y
ROLLIN: Y                                       (2.7.18)
```

2.7 CONFIGURATION QUESTION CONSIDERATIONS

The questions printed by the SYSGEN system generation program and shown in Section 2.6 concern the hardware configuration parameters and software options. Those parameters and options requiring more explanation than available in the long form of the question are explained in this section. The explanations appear in the order in which the program prints the related questions.

2.7.1 AC Power Frequency

The PDP-11 computer requires alternating current power input and is able to run on either a 6Ø Hertz (Hz) line, as is standard in the U.S.A., or a 5Ø Hertz (Hz) line which is standard in most European and Asian countries. (Hertz is the international standard of measurement for cycles per second.) The user must specify what the line frequency of his power source is.

2.7.2 Clocks

The RSTS system can operate with two types of system clocks. The KW11-L Line Time Clock divides time into intervals determined by the line frequency of the power source, either 5Ø Hz or 6Ø Hz. In areas where the line frequency of the power source is not dependable, the KW11-P Programmable Real Time Clock is recommended since its time base is derived from a crystal oscillator.

2.7.3 Power Fail Recovery Code

RSTS systems can attempt to recover from a momentary power failure by performing an automatic restart procedure. A momentary power failure is defined in Section 2.7.1 of both the PDP-11/4Ø Processor Handbook and PDP-11/45 Processor Handbook and in Section 2.2.4 of the PDP-11/20 Processor Handbook.

2.7.4 Maximum Number of Jobs

With sufficient hardware, RSTS can handle up to 16 simultaneous jobs. The user must specify the maximum number of jobs at system generation time since this parameter determines the size of several monitor tables.

The maximum number of jobs the user may configure properly and utilize efficiently depends on the memory space available and the number and type of swapping disks on the system. Memory space requirements are discussed in Chapter 1.

Jobs on the system are numbered sequentially from one to the maximum number the system can handle. Jobs include both those attached and detached.

2.7.5 Terminals

The RSTS system is designed to handle a maximum of 16 terminals. Each terminal is assigned a keyboard number ranging from Ø to 15. The console terminal is given the keyboard number Ø on all RSTS systems and is referenced by the device designator KBØ:. (Refer to Section 9.1 of the BASIC-PLUS Language Manual for the definition of device designators.)

The assignment of a keyboard number, other than that of the console terminal, is determined by the type of line interface to which the terminal is attached. The local installation can have any combination of local and remote line interfaces as long as the total number of terminal lines does not exceed 15, not including the console terminal.

The order in which SYSGEN assigns keyboard numbers is as follows: the system console terminal; all KL11, LC11, DL11A, and DL11B lines; DC11 (remote dial); and DL11E (remote dial). The answers to the configuration questions concerning the number of each type of terminal interface must accurately reflect the hardware configuration.

2.7.6 Disk Devices

Disks in the RSTS system operate in either the public or private structure. The disk which contains the system accounts and executable code of RSTS is called the system disk and is the first of the public structure. All other disks in the system are referred to collectively as non-system disks.

The most practical use of the system disk on the RSTS system is as a removable disk as opposed to a fixed head disk. If the system disk is either an RKØ5 or RKØ3 disk cartridge, it can be removed from the computer area when the system is not operating and kept in a safe place, thereby reducing the chances of inadvertent or malicious destruction. To preserve the contents of an RF11 fixed head disk, a copy must be transferred to a back up medium each time the system is shut down.

Optimum performance is obtained if the system is configured with a moving head disk (removable) and an auxiliary swapping disk (fixed head). With such a configuration, the swapping of user jobs into and out of memory is faster and more efficient. Disk accessing operations on the moving head system device can then be confined to manipulating user files and directories while the faster fixed head device takes on the burden of moving user jobs into and out of memory. In such a case, the auxiliary swapping disk acts as a logical extension of the system disk while the system is operating but contains no valuable system data when the system is not operating. At the start of time sharing operations, the initialization code creates the necessary files on the auxiliary swapping device.

2.7.7 The RF11 Disk as a Swap Only Disk

If the user indicates that both RF11 and RK11 disks are on the RSTS system, SYSGEN prints the question SWAP ONLY RF. The response to this question determines whether the user wishes to employ the fixed head disk (RF11) for swapping only and not for system use or wishes the RF11 to be used as the system disk (for swapping and system use).

Designating the RF11 disk for swapping only means that the RK11 disk is the system disk. Therefore, the RSTS Core Image Library resides on the RKØ3 or RKØ5 unit Ø. The maximum amount of space on the RF11 disk will be available for swapping only. All RF11 platters are considered a logical extension of the system disk unit DKØ: but the RF11 disk space cannot be referenced explicitly by users. To preserve user and system files, the disk cartridge can be removed from the machine when RSTS is not running.

By rejecting the RF11 for swap only operation, the user chooses the RF11 as the system disk. The Core Image Library and the swapping file (SWAP.SYS[Ø,1]) both reside on the RF11. Additionally, user files to which rapid access is necessary can be located on the RF11. All platters on the RF11 are a single logical device under RSTS (accessed by the designation DF: or DFØ:). With this configuration, less space exists for swapping. To preserve system and user files, the contents of the RF11 must be copied to a back up medium and stored in a safe place when RSTS is not running.

To select the RK11 as the system disk, type YES in response to the SWAP ONLY RF question. To select the RF11 as the system disk, type NO in response to the SWAP ONLY RF question.

2.7.8 Peripheral Devices

The use of peripheral devices in the RSTS system reduces the burden of storage requirements on the disk devices and provides a convenient means of file back-up. Program and data files that are not frequently used can be stored on magnetic tape (DECtape or magtape), paper tape (high speed, fan-folded or Teletype), and cards (marked or punched) and accessed readily when required. Media such as DECtape and magtape provide a large capacity storage back-up for critical file information.

2.7.9 Big Buffers

Big buffers are 256-word blocks of monitor memory used for DECtape operations. SYSGEN prints the BIG BUFFER question only if DECtape is on the system. On systems with DECtape devices, provide one big buffer for each DECtape drive. However, one big buffer per drive is not a definite requirement since one big buffer can accommodate any number of DECtape drives for non-simultaneous operations. Experience indicates that, unless DECtape usage is heavy, two big buffers are sufficient even for four drives. Three big buffers are recommended for six drives and four big buffers are recommended for eight DECtape drives.

2.7.10 Card Codes

If the RSTS system has a CR11 card reader, the user must configure one of three card codes. These card codes are presented for reference in Appendix D.3 of the BASIC-PLUS Language Manual. The default card code is DECØ29 code.

2.7.11 Small Buffers

The RSTS system handles transfer requests and file processing requests by means of intermediate memory storage, called small buffers. These buffers are considered a system resource, and the user must assign a sufficient number of each type at system generation time. If an insufficient number of either type is assigned, jobs running on the system can become stalled waiting until enough buffers are freed by jobs currently claiming their use.

Small buffers are 16-word blocks residing in the Monitor part of memory. The number needed by a system at any one time depends upon the dynamic requirements of the jobs on the system. The user must approximate the number of small buffers that his system can reasonably be expected to use at any one time and assign that number at system generation time.

For efficient system operation, it is recommended that at least 8 small buffers be allocated for each possible job. Thus, on a 16-user system, 128 small buffers should be available. (An indicator of good system performance is that the number of free small buffers, as reported by the SYSTAT system program, never drops below ten. Refer to the description of SYSTAT in Section 5.6 of this guide.)

On systems configured for fewer than 16 jobs, it is necessary to include more than 8 small buffers per job. For example, each active terminal requires 4 or 5 small buffers for performing input and output operations. A system having 8 terminals therefore needs between 32 and 4∅ small buffers if all terminals are to be simultaneously active. Each active job requires 2 small buffers. Thus, if the same system required that ten jobs be able to run simultaneously, 2∅ more small buffers would be needed. For each job that will run detached from all terminals, subtract the four small buffers required for terminal I/O. A running total on the 8-terminal, 1∅-job system is 52 small buffers for these two simple processing requirements.

Next, in the sample system, consider what kind of processing is necessary. One small buffer must be added for each open file on the system. If each program running on the system opens two disk files, 2∅ more small buffers must be added. If all the active programs open the maximum number of files simultaneously, 12∅ small buffers must be available. (BASIC-PLUS allows 12 open channels per user program.)

In an average system, the two file situation is much more likely, so the sample system requires 2Ø more small buffers for a total of 72.

The system requires small buffers for certain transient operations. One small buffer is used for each disk transfer queued by the monitor. On the 1Ø job, 8 terminal system, a reasonable number is approximately 2Ø more small buffers.

A line printer on the system exhausts as many as 19 available small buffers. A lower number of available small buffers places a larger burden on the system. For example, if 19 small buffers are available for use by the printer driver, the system can have 57Ø characters buffered for output to the line printer. Assume that a line printer is running at 3ØØ lines per minute (5 lines per second) and that an average line is 9Ø characters. Such a line printer empties the buffers in 1.2 seconds. A spooling program for that line printer would have to be swapped into memory every 1.2 seconds to keep the line printer running at full speed. (For a line printer running at 12ØØ lines per minute, a swap operation would be necessary every Ø.3 seconds.)

The total requirement on the 1Ø job, 8 terminal system is 111 small buffers for an average system load if the transient requirement for 1Ø is added and 19 buffers are added to handle the line printer running full speed (3ØØ lines per minute). Thus, the guideline of 8 small buffers per job is too low on such a small system. Moreover, if small buffers were subtracted from 111 to account for idle terminals and detached jobs and to allow for some slow down in line printer operations, the guideline is still inadequate. On such a system, between 12 and 14 small buffers per job is a better approximation. For larger systems having 1Ø or more jobs, eight small buffers per job is usually a good approximation. Except for occasions of heavy keyboard and line printer activity, enough free small buffers can be available to maintain good system throughput.

2.7.12 Floating Point Precision and Math Package Selection

The user can select either single precision (2-word) or double precision (4-word) floating point format for the type of numeric format to be used on his system. These floating point formats are described in Appendix F.1 of the BASIC-PLUS Language Manual.

The user can conserve memory space by omitting certain mathematical functions including SIN, COS, TAN, ATN, SQR, EXP, LOG, and LOG1Ø. These functions are described in Section 3.7 of the BASIC-PLUS Language Manual and summarized in Table 3-1 of that section.

The following list summarizes the math packages available.[1]

MA2	2-word, non-EAE, with functions
MA2E	2-word, with EAE, with functions
MA2X	2-word, non-EAE, without functions
MA2EX	2-word, with EAE, without functions
MA4	4-word, non-EAE, with functions
MA4E	4-word, with EAE, with functions
MA4X	4-word, non-EAE, without functions
MA4EX	4-word, with EAE, without functions

2.7.13 Record I/O Software

Record I/O software in a RSTS system allows a user program to perform input and output of fixed length records and enables a user to include update and PRINT USING software on the system. Record I/O is explained in Chapter 11 of the BASIC-PLUS Language Manual.

2.7.14 Update Software

If Record I/O software is configured on the system, SYSGEN asks whether the update software is to be included in the system. This software enables multiple user programs to open a file and, at the same time, update its contents provided the same physical block is not affected. Section 12.2 of the BASIC-PLUS Language Manual describes the update feature.

2.7.15 PRINT USING Option

If the user configures Record I/O software, he can include the PRINT USING optional feature in his software configuration. With PRINT USING software, BASIC-PLUS programs can perform special formatting of output as described in Section 1Ø.4.1 of the BASIC-PLUS Language Manual.

---

[1]The packages with EAE require the KE11-A Extended Arithmetic Element which is used in place of slower non-EAE software routines. The EAE unit is not the same as the KE11-E EIS option on the PDP-11/40 computer.

2.7.16 Matrix Manipulation

BASIC-PLUS can operate on an entire matrix using single statements called MAT statements as described in Chapter 7 of the BASIC-PLUS Language Manual. The user must configure this optional feature if he wants to include the matrix manipulation capability.

2.7.17 Listings

During the system generation process, the system tables, disk service and terminal service modules are assembled. Since each of the assemblies is unique to an installation, the related listings provide information valuable for documentation and maintenance purposes. If the user answers the LP11 question with YES, SYSGEN prints the LISTINGS question to allow him to have the listings printed during the system generation process. If the answer to the LP11 question is NO, SYSGEN does not print the LISTINGS query.

To print the listings at system generation time, simply type YES in response to the LISTINGS question. As a result, the listings are automatically printed later in the system generation process. The listings are quite lengthy and take approximately 3Ø minutes to print on a 3ØØ line per minute printer. To omit printing the listings, type NO to the related question.

2.7.18 Stand Alone Programs

The user can include stand alone programs in the RSTS Core Image Library (CIL). Any program included in the CIL can be bootstrapped into memory using the LOAD option at initialization time as described in Chapter 3. The DEC-supplied programs that can be included in the CIL are ROLLIN and DSKINT. ROLLIN is documented in the library document entitled PDP-11 ROLLIN Utility Program, order number DEC-11-OROAA-B-D. The document is included in the RSTS software package. DSKINT formats and initializes disks for use under RSTS and is described in Section 6.2 of this manual.

2.8 LOADING THE CIL ONTO THE RSTS SYSTEM DISK

After the user answers the configuration question concerning ROLLIN, the SYSGEN program completes building the configuration file and the second batch command file. Next, the system generation monitor executes commands in the second batch command file and generates the RSTS linked core image library (LICIL) and any listings necessary.

During the generation, messages are printed telling the user which devices to mount and how to proceed. The entire process takes between one and 3 hours depending upon the devices used and the types of listings requested. If either magtape or DECtape software is employed, the user must mount a new, formatted tape to which the RSTS LICIL is written. Instructions are printed telling the user to mount his RSTS system disk on unit Ø and showing the exact command to transfer the LICIL to the disk. If disk cartridge software is used, instructions are printed telling the user to mount a new cartridge or pack to be used as the RSTS system disk.

After the CIL is written on the RSTS system disk and the RSTS code is loaded into memory for the first time, the user must run the PATCH option described in Section 3.3.5, run the REFRESH and START options described in Section 2.10, and build the system library as described in Sections 2.11 and 2.12.

2.9 SYSLOD AND CILUS COMMAND STRINGS

This section describes the procedures and command strings employed in loading the RSTS Linked Core Image Library (LICIL) onto the RSTS system disk. All necessary instructions to perform the load operation are printed during the system generation procedure. If the user follows the standard procedure, he need not refer to the contents of this section. However, the command strings documented here are useful if the load operation is not performed as part of the system generation procedure or if the newly created LICIL must replace an old RSTS Core Image Library (CIL) on an existing system disk.

The load operation occurs as part of the final step of the system generation process. During the final step, either the user performs the load operation by executing a SYSLOD command string (magtape and DECtape distribution software) or the batch stream executes a CILUS command string (disk cartridge distribution). In both cases, the user can interrupt the process and perform the loading operation in other than the standard manner.

The final step of the system generation process for magtape and DECtape distribution software includes writing the newly created Linked Core Image Library to tape along with a copy of the SYSLOD program, the batch and configuration files, and system load maps. The SYSGEN program prints the SYSLOD command string that the user must type to load the RSTS LICIL from tape to the RSTS system disk. SYSGEN terminates by bootstrapping SYSLOD into memory. If the user preserves the tape, he can later bootstrap it to load SYSLOD and can execute the appropriate SYSLOD dialogue and command string to perform the load operation.

For disk cartridge distribution, the final step of the system generation includes loading the RSTS Core Image Library onto the RSTS system disk. The newly created Linked Core Image Library resides on the system generation disk cartridge along with the batch and configuration files, system load maps, and the CILUS program. Because the batch stream performs the loading operation, the actual command string is never printed. However, the user can terminate the batch stream when SYSGEN waits for the RSTS system disk to be mounted before the load operation begins. By preserving the system generation disk and following the procedures described in this section, the user can run CILUS to perform the load operation.

Because the user can perform the load operation using either a blank or an existing system disk, two general operations are described. Section 2.9.1 presents the procedures to load a new RSTS system onto a disk which contains no user files to be preserved. Section 2.9.2 describes guidelines for replacing an existing RSTS system on a system disk containing system and user files which must be preserved.

2.9.1 Loading the RSTS CIL Onto a Blank System Disk

The procedure and command strings detailed in this section apply only to loading a CIL onto a blank system disk. It is possible to overwrite an existing CIL on a system disk which contains system and user files without destroying the file structure. This latter procedure is described in Section 2.9.2. The procedures below destroy any existing file structure on the disk being initialized as the RSTS system disk.

2.9.1.1 DECtape and Magtape Procedures Using SYSLOD - The tape created during system generation contains a copy of the stand-alone program SYSLOD. SYSLOD is loaded from DECtape using either the BM792-YB hardware bootstrap loader or the MR11-DB Bulk Storage Loader. Refer to Section 2.2.2 for standard DECtape bootstrap procedures. SYSLOD can be bootstrapped from magtape using the MR11-DB Bulk Storage Loader. A small magtape bootstrap must be manually loaded into memory using the console switches if the configuration does not include the MR11-DB. Section 2.2.1 contains magtape bootstrap procedures. When SYSLOD is bootstrapped into memory, it identifies itself by printing the following lines.

```
SYSLOD VØ7-Ø2A
CONSOLE FILL COUNT=          Type the RETURN key
DATE: DD-MMM-YY              Type the date in format shown
DIALOGUE?

#
```

Before answering the DIALOGUE query, mount and write enable the disk to be used as the RSTS system disk. In the case of an RK system disk, an RK cartridge must be mounted on RK unit Ø. No special action is required for an RF system disk. SYSLOD will not recognize any device which is not mounted and ready when the DIALOGUE query is answered. The RETURN key is sufficient response to the DIALOGUE query. SYSLOD responds by printing the pound sign (#) when it is ready to accept a command. A single command string is sufficient to create and load the CIL onto the system disk and bootstrap the RSTS Initialization code into memory. The exact command which must be entered depends on the type of system disk. If any error messages are printed while or after the command string is entered, consult Appendix C for the proper procedure to follow. The following command strings are used for the several types of system disks.

```
DF:/NS:256:17/HO/BO<DTØ:RSTS.LCL     (To RF System Disk from DECtape)
DF:/NS:256:17/HO/BO<MTØ:RSTS.LCL     (To RF System Disk from Magtape)

DK:/NS:256:17/FO/HO/BO<DTØ:RSTS.LCL  (To RK System Disk from DECtape)
DK:/NS:256:17/FO/HO/BO<MTØ:RSTS.LCL  (To RK System Disk from Magtape)
```

Upon completion of the load operation, the RSTS Initialization code is bootstrapped into memory signalled by the printing of the OPTION query. The system manager should proceed to execute the REFRESH option according to the procedures described in Section 2.1Ø.

2.9.1.2 Disk Cartridge Procedures Using CILUS

When disk cartridge distribution software is used, the DOS program CILUS is used to load the RSTS CIL onto the system disk. CILUS will not format an RK cartridge. An RK disk should be formatted using ROLLIN prior to loading the CIL. Refer to Section 2.3.2 for procedures to load ROLLIN and format disks.

Mount the copy of the System Generation disk cartridge used for the system generation on RK unit Ø. Write-enable the drive and bootstrap the cartridge to load the DOS/BATCH monitor. Refer to Section 2.3.1 for bootstrap procedures. When the DOS monitor identifies itself, proceed as shown below to run CILUS.

```
DOS/BATCH VØ9-2Ø
DATE:  DD-MMM-YY          Type the date in format shown
TIME:  HH:MM              Type the time in format shown
DIALOGUE?  <CR>           <CR> denotes typing the RETURN key

$LO 1,1

DATE:  22-JUL-74
TIME:  12:3Ø

$RUN CILUS

CILUS VØ6-Ø7
#
```

CILUS prints the pound sign (#) when it is ready to accept a command. The CILUS command string used to create and load the RSTS CIL depends on the type of system disk. If the system is configured for an RK system disk, mount a newly formatted RK cartridge on RK unit 1. No special action is required for an RF system disk. One of the CILUS commands shown below is then used to load the CIL.

```
DF:/NS:256:17/HO/BO<DKØ:RSTS.LCL/LO   (To RF System Disk from RK cartridge)
DK1:/NS:256:17/HO<DKØ:RSTS.LCL/LO     (To RK System Disk from RK cartridge)
```

If the system disk is an RF11, the CILUS command loads the CIL and then bootstraps the RSTS initialization code into memory. The INIT code prints the system name followed by the OPTION query. The system manager should proceed to execute the REFRESH as described in Section 2.1Ø.

The command to load the CIL onto the RK cartridge mounted on unit 1 does not bootstrap the RSTS initialization code. When the load operation is complete, CILUS reprints the pound sign (#) and waits for another command. The system manager should exit from CILUS and terminate the DOS monitor as shown below.

```
#^C                         CONTROL/C exit from CILUS
.KI                         "KILL" required by DOS

$FI                         FInish for an orderly exit
TIME:  HH:MM:SS
DOS/BATCH VØ9-2Ø

$
```

The processor should be halted by moving the HALT/ENABLE switch to the HALT position. Dismount both cartridges and move the RSTS system disk to RK unit Ø. When the disk is ready, write-enable the drive and bootstrap the disk cartridge (see Section 2.3.1) to load the RSTS initialization code into memory. Proceed to Section 2.10 to refresh the new system disk.

2.9.2 Replacing the RSTS System Code

It is possible to replace the RSTS CIL on the system disk without destroying the file structure. This could be done when a new system is generated to add or change hardware support or software features. The SYSLOD or CILUS command strings used for this purpose are similar to those in the previous sections but several precautions should be taken to ensure a successful replacement. Careful adherence to these procedures is critical to avoid destroying the existing file structures (system and user files) on the system disk.

It is impossible to determine the exact size of the new CIL until it is loaded onto a disk. The first step, therefore, is to load the new monitor onto a scratch disk using the standard system generation procedures or the SYSLOD or CILUS commands described in Sections 2.9.1.1 and 2.9.1.2. The user must create the required system files on the scratch disk by using the REFRESH option described in Section 2.10.1. To determine the required size of the new Core Image Library, the user must start time sharing (START option in Section 2.1Ø.2) and execute the CATALOG [Ø,1] command which prints the length of the new CIL file CORE.SYS[Ø,1].

The next step is to determine the size of the old CIL which will be replaced. Simply obtain a directory of the system files account [Ø,1] under time sharing by using the CATALOG [Ø,1] command. The important thing to look for is the current size of the CORE.SYS file on the system disk. The REFRESH procedures of Section 2.1Ø.1 recommend that this file be made larger than the required size when the system disk is initially built. If the system manager planned for a future replacement of the system code, the current size of the old CORE.SYS file is probably larger than the required size of the new CORE.SYS file and the replacement can proceed as described in subsequent paragraphs. Otherwise, the replacement cannot be performed. All library and user files must be transferred to another disk or external medium and the system disk must be initialized (destroying the existing file structure).

Assuming the CORE.SYS[Ø,1] file on the system disk is large enough to accommodate the new CIL, the next step is to create backup copies of all library and user files on the system disk. This is a time consuming but important precaution since a typographical error or a hardware malfunction while replacing the old CIL could be disastrous. The user must perform the backup operation under time sharing using the old system. The next two sections present the SYSLOD and CILUS command strings used to replace the old CIL.

2.9.2.1 DECtape and Magtape Procedures Using SYSLOD

The tape created during system generation contains a copy of the stand-alone program SYSLOD. SYSLOD is loaded from DECtape using either the BM792-YB hardware bootstrap loader or the MR11-DB bulk storage loader. Refer to Section 2.2.2 for standard DECtape bootstrap procedures. SYSLOD can be bootstrapped from magtape using the MR11-DB bulk storage loader. A small magtape bootstrap must be manually loaded into memory using the console switches if the configuration does not include the MR11-DB. Section 2.2.1 contains magtape bootstrap procedures. When SYSLOD is bootstrapped into memory, it will identify itself by printing the following lines.

SYSLOD VØ7-Ø2A	
CONSOLE FILL COUNT=	Type the RETURN key
DATE: DD-MMM-YY	Type the date in format shown
DIALOGUE?	

```
#
```

Before the DIALOGUE query is answered, the old RSTS system disk should be mounted and write enabled. In the case of an RK system disk, the cartridge must be mounted on RK unit Ø. SYSLOD will not recognize any device which is not mounted and ready when the DIALOGUE query is answered. The RETURN key is sufficient response to the DIALOGUE query. SYSLOD responds by printing the pound sign (#) when it is ready to accept a command. The following SYSLOD commands are used to replace an old CIL on the various types of system disks.

DF:/NS:256:17/HO/BO/BL:nnn<DTØ:RSTS.LCL	(To RF System Disk from DECtape)
DF:/NS:256:17/HO/BO/BL:nnn<MTØ:RSTS.LCL	(To RF System Disk from Magtape)
DK:/NS:256:17/HO/BO/BL:nnn<DTØ:RSTS.LCL	(To RK System Disk from DECtape)
DK:/NS:256:17/HO/BO/BL:nnn<MTØ:RSTS.LCL	(To RK System Disk from Magtape)

NOTE

> In place of nnn in the /BL:nnn switch, the user must type the size of the old RSTS CIL. The switch prevents SYSLOD from loading the new CIL if it is larger than the old CIL.

The only differences between these SYSLOD commands for replacing a CIL and those used for a blank system disk are the absence of the format switch (/FO) for RK system disks and the addition of the /BL:nnn switch.

Upon completion of the load operation, the RSTS Initialization code is bootstrapped into memory signalled by the printing of the OPTION query. The system manager must reinstall all published patches using the PATCH option (see Section 3.3.5) before time sharing operations can resume with the new system. The REFRESH initialization option is not used in this case since initializing a disk destroys any existing file structures.

2.9.2.2 DECpack Procedures Using CILUS

When disk cartridge software is used, the DOS program CILUS is used to load the new CIL onto the old system disk. Since CILUS does not format an RK cartridge, the CILUS commands are nearly the same as for loading a CIL onto a blank system disk. The CILUS procedures are repeated below for continuity.

Mount the copy of the System Generation disk cartridge used for the system generation on RK unit Ø. Write-enable the drive and bootstrap the cartridge to load the DOS/BATCH Monitor. Refer to Section 2.2.2 for disk cartridge bootstrap procedures. When the DOS Monitor identifies itself, proceed as shown below to run CILUS.

```
DOS/BATCH V09-20
DATE:  DD-MMM-YY          Type the date in format shown
TIME:  HH:MM              Type the time in format shown
DIALOGUE?  <CR>           <CR> denotes typing the RETURN key

$LO 1,1

DATE:  22-JUL-74
TIME:  12:30

$RUN CILUS

CILUS V06-07

#
```

CILUS prints the pound sign (#) when it is ready to accept a command. If the system is configured for an RK system disk, mount the old RK system disk cartridge on RK unit 1. No special action is required for an RF system disk. One of the CILUS commands shown below replaces the old CIL.

```
DF:/NS:256:17/HO/BO/BL:nnn/TO:28<DK0:RSTS.LCL/LO  (To RF system disk
                                                  from RK)
DK1:/NS:256:17/HO/BO/BL:nnn/TO:28<DK0:RSTS.LCL/LO (To RK system disk
                                                  from RK)
```

NOTE

In place of nnn in the /BL:nnn switch, the user must type the size of the old RSTS CIL. The switch prevents CILUS from loading the new CIL if it is larger than the old CIL.

If the system disk is an RF11 disk, the CILUS command loads the CIL and then bootstraps the RSTS Initialization code into memory. The INIT code prints the system name followed by the OPTION query.

The command to replace the CIL on an RK system disk mounted on unit 1 does not bootstrap the RSTS Initialization code. When the load operation is complete, CILUS reprints the pound sign (#) and waits for another command. The system manager should exit from CILUS and terminate the DOS Monitor as shown below.

```
#↑C                         CONTROL/C exit from CILUS
.KI                         "KILL" required by DOS

$FI                         FInish for an orderly exit
TIME:  HH:MM:SS

DOS/BATCH VØ9-2Ø

$
```

The processor should be halted by moving the ENABLE/HALT switch to its HALT position. Dismount both cartridges and move the RSTS system disk to RK unit Ø. When the disk is ready, write-enable the drive and bootstrap the disk cartridge (see Section 3.2.1) to load the RSTS Initialization code into memory.

When the Initialization code prints the OPTION query, the system manager must reinstall all published patches using the PATCH option of the Initialization code (see Section 3.3.5). The REFRESH option is not used since initializing a disk destroys any existing file structure.

2.10 STRUCTURING THE SYSTEM DISK AND STARTING TIME SHARING

2.10.1 Refreshing the System Disk

To initialize the software structures on the system disk, the user must specify the REFRESH option. RSTS begins REFRESH with a safeguard question SURE, to which either a Y or YES answer should be given. If a Y answer is given, all subsequent refresh questions are printed in the short form indicated in the paragraphs below. If a YES answer is given, the subsequent questions are expanded to a long and more explanatory form.

RSTS proceeds to interrogate the operator about specific information needed for disk refreshing. It asks first for the date and time (24-hour time); these must be supplied in the format which the question illustrates.

RSTS asks (PACK?) for the system pack identification name and the system pack cluster size. The pack identification name is any alphanumeric name from 1 to 6 characters in length without embedded spaces. This name is for internal use only and may be whatever the system manager wishes. The allowable pack cluster sizes are 1 or 2. A cluster size of 1 allows more efficient usage of disk space but greater possible fragmentation of disk files (i.e., possible scattering of file sectors randomly across the disk surface). A cluster size of 2 maintains greater contiguity of file sectors (and hence fewer disk "seeks") but also allows greater possible waste of some disk sectors. For almost all installations, a system pack cluster size of 1 is recommended. The PACK question is answered by typing first the identification name, then a comma, and lastly the cluster size. For example, RSTS11,1↵.[1]

RSTS next asks (MNGR?) for the Master File Directory (MFD) account [1,1] password and cluster size. The password must be from 1 to 6 alphanumeric characters with no embedded spaces. This password should be kept secret, as irresponsible access to the Master File Directory can destroy the software system. With the restriction that the MFD cluster size cannot be less than the system pack cluster size, the MFD cluster size can be 1, 2, 4, 8, or 16. Large cluster sizes are more efficient with regard to minimizing disk seeks and minimizing directory accesses, but small cluster sizes are more efficient in using disk space. It is recommended that the MFD cluster size be made the same as the pack cluster size unless a large number of user accounts is required. The maximum number of user accounts possible is about 108 times the MFD clustersize. The MNGR question is answered by typing first the password, then a comma, and lastly the cluster size. For example, SECRET,1↵.

RSTS then asks (LIBR?) for the System Library [1,2] password and cluster size. The same general principles apply here as in the preceding paragraph except that the cluster size might be somewhat larger, as it affects the number and size of the files to be stored under this account. A typical answer would be SYSLIB,4↵.

RSTS next asks (SPACE?) for the number of disk blocks to be reserved in the Core Image Library to allow for possible future expansion. A practical number is 1Ø. RSTS adds the number of blocks specified by the system manager to the size of the CIL so that the CIL occupies extra space on the disk.

[1]The ↵ notation indicates that the user must type the RETURN key.

RSTS asks (BADS?) whether bad disk sectors are to be recorded. This question is included for compatibility with features planned for a future release of RSTS-11 and should at this time be answered with IGNORE↵.

RSTS asks (CR,OV,ER?) whether the CRASH.SYS file is to be created and whether the OVER.SYS and ERRM.SYS files are to be omitted. The CRASH.SYS crash-dump file, if created, occupies 28K words on the system disk and can be used by the system for recording (automatically or through operator intervention) an exact image of memory at the time of a system software failure; this is useful for reporting and analyzing system troubles which may unexpectedly appear.

The OVER.SYS overlay file, if created, is a copy of the non-resident system overlay code (7K words in length) contained in the CIL on the system disk. If the system disk is an RF-type, there is no advantage in creating the OVER.SYS file, because the system can gain fast access to this overlay code in the CIL itself. But if the system disk is not an RF-type, then the OVER.SYS file must be created (so that it will be copied from the CIL on the system cartridge to the RC fast-access disk).

The ERRM.SYS error-message file, if created, is a copy of the error message file (2K words in length) in the CIL. There is no need on any RSTS-11 configuration to create the ERRM.SYS file, since the system has equally fast access to it in the CIL; if, however, the system manager wishes to modify the error message file or to access it on line, he should create it as the ERRM.SYS file.

To create the CRASH.SYS file, answer the CR,OV,ER? question with Y; to omit it, answer with N. The OVER.SYS and ERRM.SYS files are created unless the answer to CR,OV,ER? specifically orders their omission. To omit the OVER.SYS file, follow the Y or N with a comma and an O. To omit the ERRM.SYS file, do likewise with an E. Thus

Y	causes all three files to be created,
Y,O	causes CRASH.SYS and ERRM.SYS to be created but OVER.SYS to be omitted,
N,E	causes only OVER.SYS to be created, and
N,O,E	causes all three to be omitted.

The following sample dialog shows an example of running REFRESH.

```
(line a) OPTION?  REFRESH
(line b) SURE?  YES
(line c) DD-MON-YY?  10-DEC-74
(line d) HH:MM?  19:55
(line e) PACK SERIAL ID, CLUSTER SIZE?  PACK,1
(line f) MANAGER PASSWORD, MFD CLUSTER?  MNGR,1
(line g) LIBRARY PASSWORD, UFD CLUSTER?  LIBR,4
(line h) HOW MUCH EXPANSION SPACE DO YOU WANT IN YOUR CIL?  1Ø
(line i) BAD BLOCK FILE?  IGNORE
(line j) CRASH.SYS (Y OR N), OVER.SYS (O), ERRM.SYS (E) FILES? Y,O,E

(line k) RSTS  VØ4B-17  SYS 1Ø4 TEST

(line l) OPTION?
```

Lines a, b, and i apply to all configurations and installations. If, however, the YES answer at line b is shortened to Y, REFRESH prints all subsequent questions in the short form. The responses on lines c and d should be formed, of course, according to the current date and 24-hour time.

The question on line e can be answered with whatever name is desired as a pack identification label and with a 1 or 2 pack cluster size. The questions on lines f and g can be answered with whatever passwords are desired and with cluster sizes between a minimum of the pack cluster size and a maximum of 16. In both cases the cluster sizes indicated in the lines above are recommended for average installations.

The answer to the question at line h determines the number of blocks to be added to the size of the Core Image Library. Ten blocks is an adequate number for almost all systems. Always type IGNORE in response to the question at line i.

Line j can be answered in several ways. For most installations, the following table provides the recommended answer to the CR,OV,ER? question.

Answers to CR,OV,ER? question

	system disk = RF	system disk = RK
Crash-dump file is desired	Y,O,E	Y
No crash-dump file wanted	N,O,E	N

After the system manager answers the CR,OV,ER? question, RSTS proceeds to refresh the disk by building a brand new Master File Directory [1,1], system account, [Ø,1] and System Library Account [1,2], by creating the special files specified (CRASH.SYS, OVER.SYS, ERRM.SYS), and by creating a fresh Storage Allocation Table (SATT.SYS).

After the disk has been refreshed, control returns to the beginning of the initialization code. RSTS prints its name on the terminal and asks again for an OPTION. The system manager must now start time sharing operations to build the system library as described in Section 2.1Ø.2.

2.10.2 RSTS Initialization

The system manager initializes the system for time sharing operations by using the START option. The following sample dialog shows the procedure.

```
RSTS V4B-17 SYSTEM NAME

OPTION? START                       (line a)
DD-MON-YY?  1Ø-DEC-74               (line b)
HH:MM?  2Ø:15                       (line c)
RSTS11 - SYSTEM PACK MOUNTED        (line d)
ENABLE CRASH DUMP?  Y               (line e)
CHAIN "INIT"                        (line f)
CATASTROPHIC ERROR                  (line g)
PROGRAM LOST-SORRY                  (line h)
I/O CHANNEL NOT OPEN                (line i)

READY                               (line j)

CAN'T FIND FILE OR ACCOUNT          (line k)

READY                               (line l)
```

Lines a and j apply to all installations. At line a, the system manager types START to start time sharing operations. The message at line j indicates that RSTS has completed the necessary initialization routines.

At lines b and c, the system manager types the current date and time of day in the format indicated. RSTS then executes a series of initialization routines and prints the message shown at line d to announce that it has mounted the system disk. After printing line d, RSTS executes several more initialization routines. Apparent pauses are noticeable both before and after the system prints line d.

Line e does not appear unless the system manager created the crash dump file during the REFRESH option. If the file CRASH.SYS was created, RSTS asks whether the crash-dump facility is to be enabled. RSTS accepts a Y or N answer. If the crash-dump facility is enabled and any software failure occurs, an exact image of core at the time of the failure is written onto the disk file CRASH.SYS. If the crash-dump facility is not enabled, the system never writes anything onto the CRASH.SYS file.

RSTS then concludes its initialization by printing out some messages whose meaning is not relevant here (lines f, g, h, and i). RSTS attempts to run the System Library program INIT.BAS, which as yet does not exist on the system, and hence prints lines k and l. After RSTS prints line l, the console terminal is logged into the system under the System Library account [1,2]. The system manager should proceed immediately to build the system library as described in Section 2.11.

For interpretation of initialization error messages, see Appendix E.

2.11 BUILDING THE SYSTEM LIBRARY

The source forms (.BAS) of the standard System Library programs are on the supplied medium labelled SYSTEM LIBRARY. This medium should be mounted and ready. Then all required and all optional System Library programs should be called into memory with the OLD command and COMPILEd onto disk. The required System Library programs with their protection codes are listed below:

```
LOGIN   <6Ø>
LOGOUT  <6Ø>
SYSTAT  <168>
TTYSET  <168>
REACT   <6Ø>
INIT    <6Ø>
UTILTY  <6Ø>
ERRCPY  <6Ø>
ERRDIS  <6Ø>
ERRDI1  <6Ø>
ERRCRS  <6Ø>
```

The optional System Library programs together with their protection codes are the following:

```
PIP     <4Ø>     (compiled from PIP.BAS if system has Record
                 I/O or from PIPX.BAS if system lacks Record
                 I/O.)

GRIPE   <168>
MONEY   <4Ø>
SHUTUP  <6Ø>
ODT     <4Ø>
QUOLST  <168>
SYSCAT  <6Ø>
VT5DPY  <168>
VT6DPY  <168>
ANALYS  <168>
RESEQ   <4Ø>
EDIT    <4Ø>
CONFIG
```

The types of keyboard command sequences for building the System Library are the following. For protection code <6Ø>:

```
OLD DTØ:LOGIN
READY
COMPILE
READY
```

For protection code <4Ø>:

```
OLD DTØ:PIP
READY
COMPILE <4Ø>
READY
```

For protection code <168>:

```
OLD DTØ:SYSTAT
READY
COMPILE
READY
NAME "SYSTAT.BAC" AS "SYSTAT.BAC<168>"
READY
```

The system prints READY after each command is executed. The commands show that the library files are read from the system library DECtape.

The following sections show the exact procedure to follow for each distribution medium.

2.11.1 DECtape Procedures

To build the system library from DECtape, perform the following steps.

Mount the DECtape, labelled DEC-11-ORTCB-A-UA, SYSTEM LIBRARY, on DECtape drive unit Ø.

Set the REMOTE/OFF/LOCAL switch on unit Ø to its REMOTE position.

Set the WRITE ENABLE/WRITE LOCK switch on unit Ø to its WRITE LOCK position.

Type the commands to load the source programs into memory and compile them on the system disk. The following sample dialog shows the procedure.

```
READY
OLD DTØ:LOGIN
READY
COMPILE
READY
OLD DTØ:LOGOUT
READY
COMPILE
READY
OLD DTØ:PIP              (see footnote 1)
READY
COMPILE <4Ø>
READY
```

[1] If the Record I/O feature was not included in the generation of the system software, PIP.BAS cannot be used. In such cases PIPX.BAS must be used instead. The following would be the procedure:

```
OLD DTØ:PIPX
READY
COMPILE PIP<4Ø>
READY
```

The above two steps ensure that the correct PIP is generated.

```
OLD DTØ:SYSTAT
READY
COMPILE
READY
NAME "SYSTAT.BAC" AS "SYSTAT.BAC<168>"
READY

OLD DTØ:TTYSET
READY
COMPILE
READY
NAME "TTYSET.BAC" AS "TTYSET.BAC<168>"
READY
OLD DTØ:GRIPE
READY
COMPILE
READY
NAME "GRIPE.BAC" AS "GRIPE.BAC<168>"
READY
OLD DTØ:INIT
READY
COMPILE
READY
OLD DTØ:REACT
READY
COMPILE
READY
OLD DTØ:MONEY
READY
COMPILE <4Ø>
READY
OLD DTØ:UTILTY
READY
COMPILE
READY
OLD DTØ:SHUTUP
READY
COMPILE
READY
OLD DTØ:ODT
READY
COMPILE <4Ø>
READY
```

```
OLD DTØ:QUOLST
READY
COMPILE
READY
NAME "QUOLST.BAC" AS "QUOLST.BAC<168>"
READY
OLD DTØ:SYSCAT
READY
COMPILE
READY
OLD DTØ:VT5DPY
READY
COMPILE
READY
NAME "VT5DPY.BAC" AS "VT5DPY.BAC<168>"
READY
OLD DTØ:VT6DPY
READY
COMPILE
READY
NAME "VT6DPY.BAC" AS "VT6DPY.BAC<168>"
READY
OLD DTØ:ANALYS
READY
COMPILE
READY
NAME "ANALYS.BAC" AS "ANALYS.BAC<168>"
READY
OLD DTØ:RESEQ
READY
COMPILE <4Ø>
READY
OLD DTØ:EDIT
READY
COMPILE <4Ø>
READY
OLD DTØ:CONFIG
READY
SAVE
READY
```

```
OLD DTØ:ERRDIS
READY
COMPILE
READY
OLD DTØ:ERRDI1
READY
COMPILE
READY
OLD DTØ:ERRCPY
READY
COMPILE
READY
OLD DTØ:ERRCRS
READY
COMPILE
READY
```

Proceed to Section 2.12 to establish the user accounts on the system disk.

2.11.2 Magtape Procedure

To build the system library from magtape, perform the following steps.

Ensure that the write enable ring is removed from the reel labelled DEC-11-ORTBB-A-MC9, SYSTEM GENERATION AND LIBRARY (for 9-track) or DEC-11-ORTBB-A-MC7, SYSTEM GENERATION AND LIBRARY (for 7-track).

Mount this tape on unit Ø. Ensure that no other drive is on Ø.

*

Ensure that the FILE PROT indicator is on.

Position the tape at its load point. (The LD PT indicator comes on.)

Set the ON-LINE/OFF-LINE switch to its ON-LINE position. Ensure that the READY indicator comes on.

Type the commands shown in Section 2.11.1 except replace the device designator DTØ: with MTØ: to load the source programs from the magtape. (It is assumed that the system console terminal is still logged into the system under account [1,2]. The programs are stored under account [1,2] on the magtape.) These commands load the source programs into memory and compile them on the system disk.

After compiling all the programs, proceed to Section 2.12 to create the user accounts on the system disk.

2.11.3 RKØ5 Disk Cartridge Procedures

To build the system library from an RKØ5 disk cartridge, perform the following steps.

Place the distribution disk cartridge labelled DEC-11-ORPTB-A-HA, SYSTEM LIBRARY, in RKØ5 drive unit 1.

Set the unit to RUN and WRITE ENABLE.

At the system console terminal, type the following commands to mount the disk. (The system prints the READY messages.)

```
S$ = MID(SYS(CHR$(6)+CHR$(-1Ø)+"SYSLIB"),7,4)
READY
M$ = SYS(CHR$(6)+CHR$(3)+CHR$(Ø)+SPACE$(3)+S$+
     SPACE$(12)+"DK"+CHR$(1)+CHR$(-1))
READY
```

If any errors occur, ensure that the drive unit is READY and is write enabled. Also, ensure that each line of the commands is exactly correct. If a command contains an error, retype the entire command. For information on editing BASIC-PLUS commands, see Sections 2.2.3, 3.3, and 3.6 of the RSTS-11 System User's Guide.

After the disk is logically mounted on unit 1, type the commands shown in Section 2.11.1 except replace the device designator DTØ: with DK1: to load the source programs from the cartridge. (It is assumed that the system console terminal is still logged into the system under account [1,2]. The programs are stored under account [1,2] on the disk.) These commands load the source programs into memory and compile them on the system disk.

After compiling all the programs, proceed to Section 2.12 to create the user accounts on the system disk.

2.12 CREATING ASCII TEXT AND MESSAGE FILES

After the system manager builds the system library programs, he must run the PIP system program to print sample ASCII files and to create text and message files.

2.12.1 Creating System Message Files

Two standard message files, HELP.TXT and NOTICE.TXT, would normally be established on the disk under the System Library account [1,2]. If these files are present on the disk, the system uses them; if they are not present, the system proceeds without them.

The HELP.TXT file is printed out on a user's terminal by the LOGIN program whenever a user types HELP before logging in. The intent is to inform the potential user of general information on obtaining an account, on using the system, or on logging into the system.

The system program LOGIN prints the NOTICE.TXT file on a user's terminal whenever he successfully logs into the system. NOTICE.TXT serves as a convenient means by which the system manager can inform users of recent developments, regulations, procedures, and notices.

On the distribution medium are samples of the NOTICE.TXT and HELP.TXT files which may be examined to see what the typical content of these files is. The system library PIP program may be used to create original files from keyboard input (as in the example of

NOTICE.TXT below) or to copy the samples from the distribution medium (as in the example of HELP.TXT below).

```
RUN PIP
PIP - RSTS VØ4B-17  SYSTEM TEST
#HELP.TXT<DTØ:HELP.TXT/FA
#NOTICE.TXT<KB:/FA
)

WELCOME TO RSTS-11.
PLEASE REPORT ANY PROBLEMS BY RUNNING $GRIPE.
)
↑Z

#
```

(Replace DTØ: with either MTØ: for magtape medium or DK1: for disk cartridge.)

2.12.2 Creating Control Files for INIT

Whenever the system is restarted, the initialization code logs in the console terminal (KBØ:) under account [1,2] and attempts to run the System Library program INIT. If INIT is begun after a normal system start (because the system was bootstrapped from disk), INIT uses the System Library file START.CTL to determine which actions it should perform. If INIT is begun after a system crash, it does not look at START.CTL but rather at CRASH.CTL (which is also a System Library file) to determine what it must do.

There are five types of commands which may appear in the control files; see Chapter 5 for details. The last command in each control file should be BYE, which directs INIT to log out, thus releasing the console terminal from account [1,2] or should be END if INIT runs a program such as ERRCPY which detaches from the console terminal. If the INIT program is not in the System Library or if the control file does not have the BYE command, then after a system restart the console terminal remains logged into the system under the System Library account [1,2].

The START.CTL and CRASH.CTL files may be created from keyboard input by running the System Library program PIP. Sample control files are shown below. These files are created from keyboard input through the System Library program PIP. The following sample dialogue illustrates how this is done.

```
RUN PIP
PIP - RSTS V04B-17  SYSTEM TEST
#START.CTL<KB:/FA
LOGINS
SEND RSTS-11 IS NOW ON THE AIR ...!
FORCE KB9:  ↑SET LA30S
FORCE KB4:  ↑SET VT06
FORCE KB0:  RUN $ERRCPY
END
↑Z
#CRASH.CTL<KB:/FA
LOGINS
SEND RSTS-11 RECOVERED FROM A CRASH ...!
FORCE KB9:  ↑SET LA30S
FORCE KB4:  ↑SET VT06
LOGIN KB1: [1,5]
FORCE KB1:  RUN $ANALYS
FORCE KB1:  [0,1]CRASH.SYS
FORCE KB1:  KB:
FORCE KB1:  BYE
FORCE KB1:  YES
FORCE KB0:  RUN $ERRCPY
END
↑Z
#↑Z

READY
```

2.12.3 Establishing the User Accounts on the System Disk

At refresh time, only accounts [Ø,1], [1,1], and [1,2] are created. No user accounts as yet exist. User accounts can be created one by one through the System Library program REACT, or a whole set of standard user accounts can be created by giving REACT a single command, STANDARD. In the latter case, REACT accesses a disk file named ACCT.SYS under the System Library account. Obviously, this file must be created before running REACT. A sample ACCT.SYS file is supplied on the distribution medium. The ACCT.SYS file can be created by the System Library program PIP operating from keyboard input. The ACCT.SYS file is an ASCII file, each line of which is formatted as follows:

project #, programmer #, password, disk quota, cluster size, additional identification

The disk quota is the number of disk sectors which the account is allowed to keep when logging out. A quota of Ø means no quota; i.e., that the account has no disk limit and can keep at logout all disk sectors claimed. The cluster size cannot be less than the pack cluster size; except for this restriction, the cluster size can be 1, 2, 4, 8, or 16; a cluster size of Ø indicates a default to the pack cluster size. Large cluster sizes permit very large files and/or very large numbers of files under the account but tend to waste disk space. Small cluster sizes ensure economy of disk space but tend to demand more disk accesses. Cluster sizes of 1, 2, or 4 are recommended for most systems. The additional identification field is not used by REACT but is printed out by the System Library program GRIPE. Sample lines in the ACCT.SYS file might be the following:

```
1,5,GGG,Ø,1,PRIVILEGED ACCOUNT
4Ø,4Ø,WLS,Ø,2,WILLIAM SMITH -- ACCOUNTING
5Ø,56,JJMRTN,Ø,4,JOHN MARTIN -- ENGINEERING
1ØØ,1ØØ,DEMO,1ØØ,1,GENERAL DEMONSTRATION ACCOUNT
```

The following dialogue illustrates how the ACCT.SYS file can be created from keyboard input through PIP (which is assumed to be still running from the example above).

```
#ACCT.SYS<KB:/FA
1,5,GGG,Ø,1,PRIVILEGED ACCOUNT
4Ø,4Ø,WLS,2ØØ,2,WILLIAM SMITH -- ACCOUNTING
5Ø,56,JMRTN,5ØØ,4,JOHN MARTIN -- ENGINEERING
1ØØ,1ØØ,DEMO,1ØØ,1,GENERAL DEMONSTRATION ACCOUNT
↑Z
#↑Z

READY
```

After the ACCT.SYS file has been created, REACT should be run to establish the STANDARD accounts. The dialogue for this is as follows:

```
RUN REACT
'REACT' SYSTEM ACCOUNT MANAGER
FUNCTION? STANDARD
ALL ACCOUNTS IN '$ACCT.SYS' ARE NOW ENTERED
FUNCTION? ↑C

READY
```

2.12.4 Removing the System Library Distribution Medium

After creating the ASCII files and establishing the user accounts on the system disk, the system manager should store the distribution medium and all copies and related listings in a safe place. If the distribution medium is disk cartridge, the system manager must logically dismount the system library disk cartridge as shown below.

```
RUN $UTILTY
'UTILTY' SYSTEM UTILITY PROGRAM
#DISMOUNT DK1:
#↑C

READY
```

2.12.5 Determining Remote Line Characteristics

At installations using remote hardware, specific lines of the DC11 or DL11E may be devoted to terminals that are not ASR33 compatible. The CONFIG system program would be run at this point in the system generation to permanently set the characteristics of those lines that are not ASR33 compatible. CONFIG modifies the RSTS-11 Monitor tables so that the default characteristics set up for a remote line are recognized every time the Dataset is answered by RSTS. See Chapter 5 for instructions on running the system program CONFIG.

2.12.6 Suggestions for Operation and Test

After the system disk is built, it is suggested that the user shut the system down as described in Section 5.2 and restart it to test the new START.CTL file. The user can also run DSKINT to initialize private disks or can run ROLLIN to create a back up image of an entire disk.

CHAPTER 3

HALT, START AND RESTART PROCEDURES

This chapter describes orderly halt, start, and restart/ recovery procedures (which occur in case of catastrophic error or system crash). The chapter is divided into four sections. Section 3.1 explains what an orderly halt is and what the dangers of a disorderly halt are and how to circumvent the unwanted results of a disorderly halt. Section 3.2 discusses the multiple ways of booting RSTS-11 initialization code back into core after the system has been initialized, running, and halted in such a way that operator start action is required. Section 3.3 explains the starting options available once RSTS-11 initialization code has been booted back into core. Section 3.4 explains the various causes of catastrophic errors and system crashes and the conditions which allow for automatic recovery and restart.

3.1 HALTING RSTS-11

RSTS-11 may be halted in an orderly fashion by running the System Library program SHUTUP. The program warns users that the system is about to be halted, ensures that all users log out, dismounts non-system disks, and brings the CPU to a halt at address 54. As a result, all files are properly closed and system accounting information is accurately updated. The halt leaves the program counter loaded in such fashion that depressing the CONTinue switch causes the system to be re-booted into core from its core-image library in the CORE.SYS file.

If RSTS-11 is halted by manually moving the HALT/ENABLE switch to its HALT position, clean-up operations as described above are not performed. As a result a disk storage allocation table and/or file directories are left in an obsolete state, file data can consequently become corrupted, and accounting information may be lost. The only way to recover from such a disorderly halt and to salvage possible vital file information is to raise the HALT/ENABLE switch back to its ENABLE position before any other action is taken, to depress the CONTinue switch, and thereby to return RSTS to the state in which it was before the HALT switch was used.

3.2 STARTING RSTS-11: BOOTING RSTS INTO CORE

RSTS is (re-)started by (re-)booting its initialization code into core from disk (or from the CIL DECtape). (The RSTS-11 initialization code is once-only code, loaded into high core and overlaid after its execution by the users' compiled programs.)

To (re-)boot the initialization code into core, several procedures are possible. They are described in the following paragraphs.

3.2.1 Booting RSTS from the System Disk Through the ROM Bootstrap

(a) Make sure that the system disk is mounted on disk unit #Ø. (This statement is not applicable if the system disk is an RF disk.)

(b) Make sure that the system disk is READY and WRITE-ENABLED.

(c) Move the CPU's HALT/ENABLE switch into its HALT position. Then raise it back into its ENABLE position.

(d) Set the CPU's Switch Register to 1731ØØ.

(e) Depress the LOAD ADDRESS switch; it returns automatically.

(f) Set the CPU's Switch Register to
1774Ø6 if the system disk is on RK disk unit #Ø.
or
177462 if the system disk is an RF disk.

(g) Make sure that the console Teletype is on line.

(h) Depress the CPU's START switch; it returns automatically.

This causes RSTS-11 to be booted from the disk's CORE.SYS file into core, to branch immediately to its initialization code, and consequently to print on the console Teletype the system name and request an "OPTION?" (See section 3.3.)

This method of bootstrapping is independent of any previous contents of core and requires only that the system image be the first Core Image module in the system disk's CIL (CORE.SYS) and that a special bootstrap routine reside on sector #Ø of the system disk. (These requirements are common to all methods of bootstrapping RSTS into core from disk and their fulfillment is ensured by the action taken by the RSTS-11 system itself when the system is first booted in from the CIL DECtape.) (See section 2.2.10.)

3.2.2 Re-Booting RSTS from the System Disk After a System Halt

After the SHUTUP program has brought RSTS to an orderly halt at address 54 or after a catastrophic error or system crash (see section 3.4.1) has halted RSTS at address 54, RSTS can be re-booted into core from the system disk in one of three ways.

3.2.2.1 Pressing the CONTinue Switch with Switch Register Not Set to 177777

If a system halts at address 54 and the CPU's Switch Register is not set to 177777, the operator depresses the CPU's CONTinue switch and control branches to an in-core loader. RSTS is re-booted from the disk into core in normal start-up mode. Control jumps to the initialization code, which prints the system name on the console Teletype and asks for an OPTION?. (See section 3.3.) The exact procedure to be followed for this is as follows:

(a) Check to make sure that the RUN light is off. (If the RUN light is lit, the system has not halted.)

(b) Be sure that the CPU's HALT/ENABLE switch is set to ENABLE.

(c) Make sure that the CPU's Switch Register is not set to 177777.

(d) Depress the CONTinue switch; it returns automatically.

It may be noted that starting the system from address 5Ø after a halt will produce exactly the same results as depressing the CONTinue switch.

3.2.2.2 Pressing the CONTinue Switch with Switch Register Set to 177777

If a system halts at address 54, the same procedures as described above in section 3.2.2.1 can be followed; but, at step (c) the CPU's Switch Register must be set to 177777. Control branches to the in-core loader and RSTS is booted into core in a special autorestart mode. See section 3.4.4.

3.2.2.3 Starting at Address 52 with Switch Register Set to 177777

If the system is re-started from address 52 after a system halt at address 54, three possibilities exist.

(1) If the crash-dump facility was not enabled at the previous system START-up, then (regardless of how the Switch Register is set) the system will immediately halt again at address 54.

(2) If the crash-dump facility was enabled at the last system START-up but, after starting from address 52, the system finds the CPU's Switch Register set to something other than 177777, then the system will immediately halt again at address 54.

(3) If the crash-dump facility was enabled and if after starting from address 52 the system finds the CPU's Switch Register set to 177777, then the system first writes the contents of all of core onto the CRASH.SYS file and, after completion of this writing, proceeds immediately to the in-core loader routine, causing RSTS to be re-booted into core from the system disk in the special autorestart mode described in section 3.4.4.

Clearly, the third possibility is the only meaningful one. The procedures for the third possibility are as follows:

(1) Check to make sure that the RUN light is off.

(2) Be sure that the CPU's HALT/ENABLE switch is set to ENABLE.

(3) Set the CPU's Switch Register to 52.

(4) Depress the LOAD ADDRESS switch on the CPU panel.

(5) Set the CPU's Switch Register to 177777.

(6) Depress the CPU's START switch; it returns automatically.

3.2.3 Booting RSTS from the System Disk Through the Stand-Alone Program ROLLIN

If the stand-alone program ROLLIN is in core rather than RSTS, and, if the RSTS system disk is mounted, ready, and write-enabled, then RSTS can be booted into core by giving ROLLIN the command

#/BOOT:DF (if the system disk is an RF disk)

or

#/BOOT:DK (if the system disk is on RK cartridge drive #Ø).

See Chapter 6.

After RSTS has been booted into core, it will commence with the initialization routines, printing the system name on the console Teletype and requesting an OPTION?. See section 3.3.

3.2.4 Booting RSTS from CIL DECtape Through ROM Bootstrap

RSTS can be booted into core from the preserved CIL DECtape through the ROM bootstrap by following the procedures indicated in section 2.3.14. Note, however, (as described in section 2.2.10) that this not only boots RSTS into core from DECtape but also loads the entire CIL from DECtape onto the system disk and then re-boots RSTS (which is the first core image on the disk CIL) from disk into core. This procedure can be used to restore the CIL to the disk in case some operational error has corrupted it.

3.3 STARTING RSTS-11: STARTING OPTION

Whenever RSTS is (re-)booted into core in such fashion as to take the normal start-up mode, control always jumps to the beginning of the initialization code and the system asks for an OPTION?. At this point, the operator must type in on the console keyboard one of the valid options listed below:

Options at Start-up time

long form of operator response	short form of operator response
START	S or LINE FEED
REFRESH	R
LOAD	L
BOOT	B

If anything other than a valid option is typed by the operator, the system rejects it, prints the valid options, and again requests an "OPTION?". If the operator types a valid option, the system proceeds according to the details explained below.

3.3.1 Normal System START-up Option

If the operator responds to OPTION? with START or LINE FEED, the system executes its normal initialization code, which, after interrogating the operator, puts RSTS into its full running state. The dialogue involved in this option appears in the following format:

```
RSTS V4A-12 SYSTEM #213

OPTION? START                                  (line a)
DD-MON-YY? 11-AUG-72                           (line b)
HH:MM? 15:14                                   (line c)
V4A12 - SYSTEM PACK MOUNTED                    (line d)
ENABLE CRASH DUMP? YES                         (line e)
CHAIN "INIT"                                   (line f)
CATASTROPHIC ERROR                             (line g)
PROGRAM LOST-SORRY                             (line h)

READY                                          (line i)

SYSTEM INITIALIZATION PROGRAM                  (line j)

BYE$FAST$                                      (line k)
READY                                          (line l)

CONFIRM:  SAVED ALL DISK FILES; 7Ø1 BLOCKS IN USE  (line m)
JOB 1 USER 1,2 LOGGED OFF KBØ AT 11-AUG-72 Ø3:14 PM
SYSTEM RSTS V4A-12 SYSTEM #213
RUN TIME WAS .5 SECONDS
ELAPSED TIME WAS 19 SECONDS
GOOD AFTERNOON
```

Lines a and j are the same for all installations. The questions on lines b and c are answered according to the current date and time. Pauses occur before and after line d is printed when the system executes several initialization routines.

Line e does not appear unless the crash-dump file CRASH.SYS was created at disk Refresh time. (See sections 2.3.15 and 2.2.12.) Answering Y to the question at line e enables the crash-dump facility; answering N disables it; the enabled or disabled condition prevails until the next system start-up. (See section 3.4.3.)

Lines g, h, and i have no real significance to the operator in this context. They are caused by a deliberate system error-simulation in order to take advantage of standard error-handling routines which effect a clean initialization of Job 1 on KBØ: under account [1,2].

The initialization code also forces a CHAIN INIT statement into Job 1's keyboard input buffer. This is pre-echoed as line f and causes Job 1 to begin running the System Library program INIT.

The INIT program name is printed at line j, performs the operations which the START.CTL file (see Section 5.1) directs it to do, and exits back to the System Monitor upon executing its own END statement. The exit to the system Monitor causes the READY at line 1.

If the START.CTL file contains a BYE command, the INIT program forces the character-string "BYE<ALT>FAST<ALT>" into the Job 1 keyboard input buffer before exiting back to the System Monitor. This echoes as line k. When the System Monitor detects this string in the input buffer, it causes the console terminal (KBØ:) to be logged off from account [1,2] and causes the print-out at lines m and following. If the START.CTL file contains no BYE command, the console terminal remains logged in under account [1,2] and lines k, m, and following do not appear. In either case the RSTS-11 system is fully initialized and running.

At this point the operator must decide whether the automatic restart facility is to be enabled or to be left disabled. (See Section 3.4.3.) If the automatic restart facility is to be enabled, the CPU's Switch Register must be set to 177777; (i.e., all switches in the up position); the automatic restart facility remains enabled as long as the CPU's Switch Register remains set to 177777. If a Switch Register toggle between positions 15 and Ø is set down to its Ø position, the automatic restart facility is disabled for the duration of this condition.

3.3.2 Disk-REFRESHment Option

The OPTION question should be answered with REFRESH or R only if the system-disk structure is to be completely rebuilt from scratch. If this option is chosen, all accounts and all files existing on the public disk structure are lost.

The details of disk refreshment are explained in section 2.1Ø.1. All of those details are directly applicable here. One further option, however, exists for the case where an already structured RSTS system disk is to be refreshed. If the SURE question is answered with O (for old) rather than with Y or YES, then the disk-refreshing routine asks the operator only for the date and time and extracts from the pre-existing system disk the answers which it needs for the questions (which therefore are not printed out on the terminal). As a result, the refreshed disk is built anew with a fresh CRASH.SYS file only if a CRASH.SYS file existed previously on that disk, with a fresh OVER.SYS file only if an OVER.SYS file existed previously on that disk, and with a fresh ERRM.SYS file only if an ERRM.SYS file existed previously on that disk. Similarly, the refreshed disk is given the same pack identification label, the same [1,1] password, the same [1,2] password, and the same respective cluster sizes as it had before. The BAD.SYS file (reserved for future system implementation) contains the same information which it contained previously.

An alternative exists to the above-described all-inclusive use of the O (old) answer. Instead of answering the SURE? question with O, the SURE question can be answered with Y or YES. The system then proceeds to ask the operator all the individual questions described in Section 2.1Ø.1. For any individual answer, however, O can be typed in on the keyboard. This causes the system to fetch the answer for that individual question from the pre-existing system-disk structure.

At the conclusion of disk-refreshment, control returns to the beginning of the initialization code, and the OPTION? question is again asked.

3.3.3 LOAD Option

If the initialization code OPTION? question is answered with LOAD or L, then the system asks for the name of the program to be loaded from the CIL (LOAD PROGRAM). The operator then types in the name of one of the programs optionally included at the end of the CIL (see Section 2.7.18).

Upon reception of the operator's response, the system searches the disk for the desired program and loads it into memory and starts its execution. (See Chapter 6 for details on DSKINT and ROLLIN and note that LOADing either of these programs into memory overlays the RSTS Monitor.) If the system cannot find the requested program in the disk's CIL, it prints the message PROGRAM NOT FOUND and then returns to the OPTION question at the beginning of the initialization code.

3.3.4 BOOT Option

If the OPTION question is answered with BOOT or B, the system asks from what device (BOOT DEVICE:) a program is to be bootstrapped. The operator must then type in a device-type; see the table below.

Operator's response to BOOT DEVICE:	Device from which a program is bootstrapped into core
DF	RF11/RS11 disk
DK	RK disk cartridge drive #Ø
DC	RC11/RS64 disk
DP	RP disk pack drive #Ø
DT	DECtape drive #Ø
MT	Magtape drive #Ø

Upon receiving the operator's response, the subsequent code determines whether the indicated device exists on the system. If it does not exist, an error message NOT A VALID DEVICE will be reported. If the device does exist, the system branches to the proper internal bootstrap code, which reads the first 64 words from block #Ø on the device into the first 64 words of memory and transfers control to address Ø. By this means, for example, the DOS Monitor could be bootstrapped into memory from its disk (and overlay the RSTS Monitor).

3.3.5 PATCH Option

The RSTS Initialization code PATCH option provides a convenient means for altering the RSTS system code as errors are found and corrections are published. When a RSTS system generation is performed, all patches are installed immediately after the Core Image Library (CIL) is loaded onto the system disk. This is necessary since patches may affect the initialization code used to build required file structures, create the system files, and set up tables used during normal timesharing. Patches are published in the RSTS Installation Notes if problems are uncovered after a "code freeze", but before a new release is available from Digital's Software Distribution Center. Thereafter, patches are published in the DIGITAL Software News.

The PATCH option makes permanent changes to the RSTS CIL on the system disk. The CIL is made up of several modules including SY (the resident monitor and device drivers), OV (Overlay Code) and ER (error messages). Any of these modules may be altered using the PATCH option.

Patches take many different forms. Some are in-place patches to one or more words in one or more modules. Others require patch space in the affected modules. In some cases, patches affect fixed addresses and are straightforward; however in most cases it is necessary to refer to the system load map to find the addresses of affected sections. Published patches describe the procedures required to make the alteration correctly.

The PATCH option is called by typing PATCH or simply P in response to the initialization code OPTION query. PATCH replies by asking for a MODULE NAME (one of the three listed above), a BASE address, and an OFFSET address. The module name determines the CIL module to be changed. The response "SY" indicates that the patch applies to the resident monitor. The base address further determines the actual locations to be patched. For example, the base address for the PRINT USING section of BASIC-PLUS is found in the load map and might be entered as the response to the BASE query. Finally, the offset address is the first location to be changed relative to the specified base. For example, a PRINT-USING patch may begin at an offset of 1ØØ octal bytes from the beginning of Print Using. After these items are entered, PATCH prints the old contents of the specified location and opens the word for change. PATCH opens and changes successive locations depending on the user responses.

Details for the use of the patch option are included in the example presented below.

```
RSTS VØ4B-17 SYS 1Ø4 TEST

OPTION?  PATCH
MODULE NAME?  SY
BASE?  122722
OFFSET?  Ø
MODULE    BASE      OFFSET    OLD        NEW
SY        122722    ØØØØØØ    ØØØØØØ?    16115
SY        122722    ØØØØØ2    ØØØØØØ?    14
SY        122722    ØØØØØ4    ØØØØØØ?    11561
SY        122722    ØØØØØ6    ØØØØØØ?    177776
SY        122722    ØØØØ1Ø    ØØØØØØ?    2Ø7
SY        122722    ØØØØ12    ØØØØØØ?    ↑C

RSTS VØ4B-17 SYS 1Ø4 TEST

OPTION?  BOOT
BOOT DEVICE:  DF

RSTS VØ4B-17 SYS 1Ø4 TEST

OPTION?
```

All numbers printed by the PATCH option and all numeric responses are octal. In the example, the number 122722 indicates an address found in a load map or computed. PATCH does not perform any arithmetic; hence, expressions of the form [NAME] + 2Ø must be manually calculated using 2's complement arithmetic. (If unfamiliar with the octal representation of binary numbers or with 2's complement arithmetic, consult a Software Support Representative.) As PATCH opens successive locations, it prints the current or old location contents and waits for new data to be entered as an octal word. A carriage return (CR) is used to enter the new data. PATCH then sequences to the next location. A line feed <LF> with no new data causes PATCH to sequence to the next location without altering the current location. PATCH continues to open successive locations until the CONTROL/C combination is typed. CTRL/C returns to the initialization code OPTION query.

Note that changes are made immediately upon typing the carriage return key. If an error is made it becomes necessary to reenter the PATCH option to correct the mistake. Printing the old contents of a location provides a check for proper placement of a patch. If the old contents of any location shown in a published patch are not identical to those printed by the PATCH option, all locations should be restored to their old contents. This may indicate an error in the use of load maps or an error in the published patch itself. Finally, a complete patch may be double checked by reentering the PATCH option and using the line feed key to examine successive locations.

NOTE

Whenever the user patches the Resident Monitor (module SY), he must use the BOOT option to load the patched version from disk into memory.

3.4 AUTOMATIC RECOVERY AND RESTART FACILITIES FOR HANDLING CATASTROPHIC ERRORS AND SYSTEM CRASHES

3.4.1 Nature and Causes of Catastrophic Errors and System Crashes

A catastrophic error or a system crash is an error-trap to vector 4 or vector 1Ø. (See section on Processor Traps in PDP-11/2Ø Processor Handbook 2.2.4 or in PDP11/45 Processor Handbook 2.7. See Section 3.4.2 below for distinction between catastrophic error and system crash.) Such traps can be caused, for example, by referencing a nonexistent (or non-responding) Unibus address (bus time-out trap), by referencing an odd address with an instruction that requires a word-address, or by attempting to execute a reserved or nonexistent instruction.

Catastrophic errors and system crashes can therefore be due to any of four types of problems: (a) configuration errors, (b) privileged-account programming errors, (c) hardware malfunctions, and (d) system software deficiencies. Each of these is discussed individually below.

3.4.1.1 Configuration Errors - If the software configuration and the hardware configuration do not correspond exactly, bus time-out traps occur whenever the software attempts to address a peripheral interface which, for the software, logically exists but which for the hardware, does not physically exist. Thus at system-generation time, it is imperative that the operator respond to SYSGEN's questions about peripherals with answers which accurately describe the existing hardware configuration.

3.4.1.2 Privileged-Account Programming Errors - The RSTS system software is designed to protect itself against programming errors perpetrated under non-privileged accounts. The system itself, upon detecting such an error, aborts execution of the would-be error and reports a corresponding error message to the guilty user.

The RSTS software is vulnerable to certain types of errors perpetrated under privileged accounts. By intent and design, the system manager (and those to whom he assigns any [1,*] account) have been given extensive powers which are not available under non-privileged accounts. These powers allow privileged users to modify System Library programs or to create utility programs in such fashion that they can access and modify any part of core. Extreme care must be used when programming with privileged SYS functions. A mistake can cause the system to take an error trap.

3.4.1.3 Hardware Malfunctions - Hardware malfunctions can be responsible for crashing the system. If unexplainable and random-type system crashes or catastrophic errors occur (particularly on systems which hitherto have been functioning well), it is likely that a hardware problem has arisen. In such cases a DEC Field Service Representative should be contacted.

3.4.1.4 System Software Deficiencies

Although every attempt has been made to detect and eliminate system software bugs, the paths and routes which control can take through the RSTS software are incalculably numerous. It is possible that, given certain conditions and certain sequences, the RSTS software can trap to vector 4 or vector 1Ø. If a problem of this type is discovered (it should be reproducible in a defined environment and under defined conditions), a DEC Software Specialist should be contacted. If new problems of this type become known, DEC reports them in The Digital Software News. (Known problems are reported in Chapter 7.)

3.4.2 Automatic Recovery from Catastrophic Errors

At system-generation time, SYSGEN asks whether the module for automatic recovery from catastrophic errors is to be included in the system. If a Y answer is given, the system is configured with the NF (No-Fail) module; otherwise the system is configured without it.

If the system is configured with the No-Fail module, then whenever a trap occurs to vector 4 or vector 1Ø, the system distinguishes the trap as one of two categories: it is either (a) a catastrophic error for which one particular user is responsible or (b) a system crash for which some Monitor bug is possibly responsible. If the system is configured without the No-Fail option, this distinction is not made; instead all traps to vectors 4 or 1Ø are treated as system crashes, regardless of their true cause. The handling of system crashes is treated below in section 3.4.3. The handling of catastrophic errors by the No-Fail code is as follows.

The system determines which user was responsible for the error-trap. It flags that user's job with a special code which will cause the system to reinitialize that user's job area completely when it is next his turn to run. The system prints out on that user's terminal the message CATASTROPHIC ERROR PROGRAM LOST-SORRY.

At this point the system determines whether the crash-dump facility was enabled at system START-up time (which is possible only when the system has the CRASH.SYS file created at system-disk Refresh-time). If the crash-dump facility was enabled, the system copies the contents of core onto the CRASH.SYS file and resumes normal time-sharing operations. If, on the other hand, the crash-dump facility was not enabled, the system resumes its normal time-sharing operations. The reinitialized user is in the same state as he would be if he had just logged in and received the NEW OR OLD question.

A single catastrophic error does not affect any user other than the one whom the system considers guilty, and does not halt the system. If, however, two catastrophic errors occur within the same minute (more accurately stated, two error-traps within the same minute), the system halts at address 54. This protects the system against an infinite loop of error-traps should they be caused by some repeating hardware malfunction.

See Appendix F for a flow-chart portraying the design of the No-Fail automatic recovery process.

3.4.3 Automatic Restart after System Crashes

If a system crash is distinguished by the No-Fail code (see section 3.4.2) or if any error-trap occurs in a system configured without the No-Fail code, the system takes an automatic restart path only if both of two conditions are fulfilled:

(a) the crash-dump facility must have been enabled at system START-up time (possible only when the CRASH.SYS file exists) and

(b) the CPU's Switch Register must currently be set to 177777.

If either condition is not fulfilled (or if both are not fulfilled), the system does not take the automatic restart path but simply halts at address 54.

If the system halts at address 54, the operator may choose one of three procedures.

(a) He depresses the CPU's CONTinue switch, which causes the system to be re-booted into normal system start-up mode.

(b) He sets the CPU's Switch Register to 177777 and depresses the CPU's CONTinue switch, this causes the system to be re-booted from disk and to be started in the special autorestart mode described below.

(c) Finally, the operator starts the CPU at address 52 with CPU's Switch Register set to 177777 (see section 3.2.2.3). This causes the system first to write the contents of core onto the CRASH.SYS file (provided the crash-dump facility had been enabled) and then to be re-booted from disk into the special auto-restart mode described below.

If the system takes the automatic restart path, no halt occurs. Instead, the system first writes the contents of all of core onto the CRASH.SYS file and then re-boots itself into core from the system disk. After the system has been re-booted into core, control jumps to the initialization routines. At this point the system recognizes the fact that it was not activated through a normal system start-up but rather through an automatic restart and consequently initializes itself in auto-restart mode.

3.4.4 Auto-restart Mode Initialization

When the system is initialized in auto-restart mode, control by-passes all parts of the initialization code which call for operator intervention (e.g. questions concerning the OPTION?, date, time, etc.) and initializes the system using information already stored in core. The system logs Job 1 in on KBØ: under account [1,2] and causes it to run the System Library program INIT beginning at line 1ØØ. Since, in auto-restart mode, INIT begins at line 1ØØ (rather than at its lowest line number), it takes directions from the CRASH.CTL System Library file (rather than from the START.CTL file). The CRASH.CTL file should direct INIT to perform all operations which the system manager considers necessary in the case of an automatic restart (e.g., to send an appropriate message to all terminals), and it should end with a BYE directive. If this is done before INIT exits to the system Monitor (as a result of its END statement), it will force a character-string (See section 3.3.1) into KBØ:'s input buffer (echoed on the terminal) which causes the system to log Job 1 off from KBØ: and account [1,2]. At this point the system is in its normal time-sharing mode.

The print-out which appears on the console Teletype is similar in format to the following. Section 3.3.1 explains why most of these lines appear. (The I/O CHANNEL NOT OPEN line may or may not appear, depending on a random state of a word in core.)

```
SYSTEM RELOADING   !!!


V4A12 - SYSTEM PACK MOUNTED
CHAIN "INIT" 1ØØ
CATASTROPHIC ERROR
PROGRAM LOST-SORRY
I/O CHANNEL NOT OPEN

READY


SYSTEM INITIALIZATION PROGRAM

BYE$FAST$
READY

CONFIRM:  SAVED ALL DISK FILES; 7Ø1 BLOCKS IN USE
JOB 1 USER 1,2 LOGGED OFF KBØ AT 14-AUG-72 Ø6:47 AM
SYSTEM RSTS V4A-12 SYSTEM #213
RUN TIME WAS .8 SECONDS
ELAPSED TIME WAS 27 SECONDS
GOOD MORNING
```

As stated at the end of section 3.4.2, two error-traps within the same minute (be they catastrophic error or system crash) halt the system at address 54.

See Appendix F for a flow-chart of the automatic restart logic.

CHAPTER 4

SYSTEM ACCOUNTS, PRIVILEGE, AND SYSTEM FUNCTION CALLS

4.1 SYSTEM ACCOUNTS

Any RSTS-11 system has three system accounts which are integral to the operation of the system. The MFD account [1,1] is used on the system device and other disk devices in the system to control access and file creation. It is described in Section 4.1.1. The system library account [1,2] is used by the RSTS-11 system to manage a library of generally available and restricted use system programs and message and control files. It is described in Section 4.1.2. The system account [0,1] contains RSTS-11 Monitor files and routines which are critical to the operation of the system. System account [0,1] is described in Section 4.1.3.

4.1.1 MFD Account [1,1] on the System Device

Access to the RSTS-11 system is controlled by the use of project-programmer numbers and passwords. The system manager, operating under his privileged account, creates a new account by the system program REACT. (See Chapter 5 for a description of REACT.) The project-programmer number and password of the new account are given, along with other information, to allow a user the means to access system facilities.

When a new account is created, the new account information is stored on the system device under the MFD account [1,1]. The password is stored in packed Radix-50 format. When the new user first creates a file, an area is created on the system device which is related directly to the user's account (project-programmer number). This area is called the User File Directory or UFD. The UFD contains information concerning the files created under that account number.

The account [1,1] contains a catalog of information of all User File Directories on the system, called the Master File Directory or MFD. When a user attempts to gain access to the RSTS-11 system by giving his account and password, the system program LOGIN is run automatically. LOGIN checks the MFD on the system device to determine whether the account number and password given compare with one stored in the MFD. If so, the user is allowed access to the system.

Other account information is stored in the MFD for each account in the system. This information is summarized in Table 4-1.

The account information in the MFD is accessed by various system programs. The LOGIN system program has already been mentioned. The MONEY system program references the accumulated usage information. The system manager uses the MONEY system program to read and reset usage information. The disk storage information is referenced by the LOGOUT system program. The system manager can change the quota by use of the UTILTY system program. System access information is checked by the LOGIN system to determine the presence of a detached job. Refer to the descriptions of the individual system programs in Chapter 5 for more exact details.

A facility is provided whereby the system manager can write programs which access the information in the MFD. See the description of the system function SYS in Section 4.3.

4.1.1.1 <u>MFD on Other Devices</u> - The system device exists in what is called the public structure. See Chapter 12 of the <u>BASIC-PLUS Language Manual</u> for a discussion of the public and private structures. Additional disk devices can be added to the public structure and to what is called the private structure. Each device that is added to the system also contains its own MFD. The MFD of each additional device is created when the system manager runs the stand-alone program DSKINT. See Chapter 6 for the description of DSKINT. The MFD of any device contains account and storage information for that device only.

The MFD is treated differently on devices other than the system device. The RSTS-11 system allocates space for a user's file on the device in the public structure which has the most free space. If the user's account is not in the MFD of that device which has the most free space, his account information is added dynamically into the MFD of that device and a UFD is created for that user.

The MFD on a device within the private structure is different. If a user desires to create a file on a disk within the private

Table 4-1

Account Information Stored in the MFD on the System Device

Type	Description	Explanation
Identification	Project-programmer number (account)	Refer to Chapter 9 of BASIC-PLUS Language Manual.
	Password	6 characters stored within 2 words in Radix-50 format.
Accumulated Usage	Central Processor Unit (CPU) time (Run Time)	Number of hours, minutes, and seconds to the tenth of a second (hh:mm:ss.t) of processor time the account has used to date.
	Connect time (login time)	Number of hours and minutes (hh:mm) the user has been connected to the system via a terminal or remote line.
	Kilo-core-ticks (kct's)	Core usage factor. One kct is the usage of 1K words of core for one tenth of a second.
	Device time	Number of hours and minutes of peripheral device time the account has used.
Disk Storage	Quota	Number of 256-word blocks the user is allowed to retain at logout time.
System Access	Logins	Number of times the user has logged in.
	Logouts	Number of times the user has logged out.

structure, he cannot gain access if his account information does not already exist on the MFD of that device. The system manager or a privileged user grants WRITE access to a device in the private structure by entering the account information with the REACT system program which is described in Chapter 5.

The MFD on devices within the public and private structure aside from the system device contains its own functional information. The information is pack label and account information.

The pack label information consists of pack cluster size, pack status (private or public), and pack identification (id). The pack id is the 6-character name in RADIX-50 format given at the time the disk was initialized via the stand-alone program DSKINT. The pack id is utilized whenever the disk is logically mounted via the system program UTILTY or via INIT. (A distinction must be made here between physical mounting and logical mounting. The disk is mounted physically by making the hardware ready to use the disk. The disk must be mounted logically by the system program UTILTY or from commands in START.CTL or CRASH.CTL by the INIT system program to enable the software to recognize and access the disk. The disk must also be logically dismounted before it is physically dismounted if corrupted disk structures are to be avoided.)

The MFD contains accounting information much the same as that for each account in the MFD of the system device as described in Table 4.1. Also included are the device time and UFD cluster factor for the account.

4.1.2 System Library Account [1,2]

The system library account [1,2] is created on the system device during the REFRESH operation of the system build procedures. During subsequent operations of the system build procedures, the system manager creates the contents of the system library account [1,2]. This section describes the contents of account [1,2]. Refer to Chapter 2 of this guide for a description of how the system library is built.

The system library catalogs, as files, system programs which are available to general users and to privileged users. It also contains text and control files used by system programs. Refer to Chapter 2 of the RSTS-11 System User's Guide for a user-level description of system library files.

The contents of the system library are presented in Table 4-2. For a more detailed description of the contents, refer to Chapter 4 of the RSTS-11 System User's Guide or Chapter 5 of this guide under the section explaining the related program.

For operational purposes, the system library is accessed automatically during normal system start-up. As a result of specifying the START option when RSTS-11 is bootstrapped into core, the console keyboard is logged in automatically under account [1,2]. (See Section 3.3 of this guide for the description of the START option.) At this time, the commands in the file START.CTL, stored under account [1,2], are executed by the INIT system program, which is also stored under account [1,2]. Unless the system manager desires to add to or modify the contents of the system library after the system has been built, he need not log in under account [1,2].

For auto-restart purposes, the system library file CRASH.CTL is used by the INIT system program when recovering from system crashes. When auto-restart mode is entered, the RSTS-11 system performs actions similar to those described above for normal start-up, except that the contents of the system library file CRASH.CTL are executed. (See the description of auto-restart mode initialization in Section 3.4.4.)

Two files, HELP.TXT and NOTICE.TXT, are provided to supply information to the user. The HELP.TXT file is printed automatically by the system program LOGIN if a user types HELP at a logged-out terminal. The NOTICE.TXT file is printed automatically by the system program LOGIN after a user has typed a valid account number and password during the log-in procedure. The NOTICE.TXT file provides a means by which the system manager can communicate up-to-date installation information to users. One other system library information file, PIP.TXT, is printed out when the user types HELP while running the related system program.

Table 4-2

File Contents of System Library

File Name	Status	Origin	Related Programs	Description
Individual System Program Names	Both required and optional	Compiled by manager from System Programs supplied with distribution kit.		See Chapter 2 System Generation Procedures, "Building the System Library"
HELP.TXT	optional	Created by manager via PIP during system generation	LOGIN	System message file which contains general information about system use
NOTICE.TXT	optional	Created by manager via PIP	LOGIN	System message file which contains system information of daily interest; updateable by system manager.
ACCT.SYS	required (if the STANDARD function of the program is used.)	Created by manager at system generation time via PIP	REACT, GRIPE	Standard account file used by REACT at system generation time to facilitate automatic creation of user accounts and by GRIPE program to identify user making the comment.
START.CTL	required	Created by manager via PIP	INIT	Normal system start-up control file for INIT. Refer to the description of INIT in Chapter 5 for the contents of this file.
CRASH.CTL	required	Created by manager via PIP	INIT	Auto-restart mode control file for INIT. Refer to the description of INIT in Chapter 5 for the contents of this file.
GRIPE.TXT	optional	Dynamic creation and deletion on an as needed basis	GRIPE	Retains comments submitted by a user via GRIPE for inspection by the system manager. See Section 5.9 for a description of GRIPE.
PIP.TXT	accompanies the related system program	Created by transferring the file from the RSTS-11 kit to the system library using PIP	PIP	System program file which contains operating instructions of the related system program; printed in response to HELP while running the related system program.

The file GRIPE.TXT is created dynamically when a user runs the GRIPE system program. The system manager has read and delete access to the GRIPE.TXT file by means of the GRIPE system program.

4.1.3 System Account [0,1] on the System Device

The system account [0,1], like the system library account, is created on the system device during the REFRESH operation of the system build procedures. Account [0,1] has a password identical to the pack id and it contains name and retrieval information for system files. Account [0,1] is used solely by the RSTS-11 Monitor to execute various operational and control actions. In this respect, it is a direct access account for use by the RSTS-11 Monitor, and no user need be concerned with referencing its contents. The account can be used by a software specialist to make patches in the RSTS-11 system. This section describes the contents of the account in order to familiarize the system manager with the internal operations of the RSTS-11 system.

The contents of account [0,1] are presented in Table 4-3. The file SATT.SYS is the mechanism by which RSTS-11 controls the allocation and deallocation of disk storage space for the disk. The file maps the entire space on the disk in a bit map table called a SAT (Storage Allocation Table). Each bit in a SAT represents an occupied disk cluster; each bit cleared in a SAT represents a free disk cluster. Disk storage is typically allocated in multiples of the pack cluster size. A cluster represents a fixed number of 256-word blocks, the number of 256-word blocks being determined by the cluster size. The blocks within a cluster are always contiguous, whereas clusters may be scattered across the surface of the disk.

Cluster sizes are defined for disk devices (packs), directories, and files. Table 4-4 presents the types of clusters, the range of cluster sizes for each type, and the time when the sizes are defined. The pack cluster size determines how many contiguous 256-word blocks are represented by each bit in a SAT on a disk, and also determines the minimum cluster size for directories and files on the disk. The directory cluster size determines the size to which a directory can expand, given the fact that a directory, whether an MFD or UFD, can occupy only seven clusters. The file

Table 4-3

Contents of System Account [0,1]

File Name	Location	Status	Description
SATT.SYS	Each disk device in system	Required	Storage allocation file containing retrieval and structure information for each disk device.
ERRM.SYS	System or swapping disk only	Optional (except on systems with a swapping disk)	Copy of error messages in CIL (CORE.SYS) which can be accessed and modified by system manager.
OVER.SYS	System or swapping disk only	Optional (except on systems with a swapping disk)	Copy of non-resident (overlay) code in CIL (CORE.SYS).
BUFF.SYS	System or swapping disk only	Required for DECtape	File to retain DECtape directories.
SWAP.SYS	System or swapping disk only	Required	File used to store user job images not in core.
CRASH.SYS	System disk only	Optional	File used to store exact image of core following a system crash.
CORE.SYS	System disk only	Required	RSTS-11 Core Image Library containing resident and non-resident (overlay) code, symbolic definitions, and error messages.
BADB.SYS	(not implemented but appears on each disk drive in the system)		File which catalogs bad blocks on disk.

Table 4-4

Range of Cluster Factors

Cluster Type	Minimum Size	Maximum Size	When Defined
Pack (for system disk)	1	2	At REFRESH time (stored in MFD)
Pack (for non-system disk, public and private)	1	16_{10}	At initialization time via DSKINT stand-alone program (stored in MFD)
Directory (both for MFD and UFD)	Pack cluster size	16_{10}	At creation of the directory via either REFRESH, DSKINT, REACT, or SYS system function.
File	Pack cluster size	256_{10}	At creation of the file via either an OPEN or OPEN FOR OUTPUT.

cluster size is discussed extensively under the heading "CLUSTERSIZE Option" in Chapter 9 of the BASIC-PLUS Language Manual.

The files SWAP.SYS, OVER.SYS, and BUFF.SYS are crucial to RSTS-11 system response time. These files should reside on a fast access disk. If the system disk is not a fast access type (i.e., not an RF type), a copy of the non-resident (overlay) code is taken from the RSTS-11 CIL (CORE.SYS), placed in the file OVER.SYS and stored on the fast access disk, if available. The file SWAP.SYS keeps all jobs that are temporarily transferred out of core.

DECtape processing is expedited by the use of BUFF.SYS. When a file on DECtape is opened, the directory of the DECtape is read into the file BUFF.SYS. Any updates to the DECtape directory which arise during processing cause the copy on the fast access disk to be manipulated. This technique eliminates the need for continuous winding and rewinding of DECtape. The copy of the DECtape directory in BUFF.SYS is read back to the DECtape when the last file open on the DECtape unit is closed or any output file is closed.

The copy of the BASIC-PLUS error messages from the CIL (CORE.SYS) can be kept in the file ERRM.SYS, which can be accessed and modified.

The RSTS-11 Core Image Library is contained in the file CORE.SYS on the system disk. This file contains all executable code for system operations, resident and non-resident, and optionally contains code for stand-alone programs such as DSKINT and ROLLIN. The CRASH.SYS file can be created to retain a copy of core at the time of a system crash. (See Chapter 3 and Chapter 5 for discussion of system crashes and analysis of system crashes, respectively.)

4.1.3.1 System Account [0,1] on Non-System Disks - The system account [0,1] on non-system disks contains pointers to system routines on the system disk, a BADB.SYS file, and a SATT.SYS file for recording the used space on the disk. The system account on non-system disks is created by the disk initializer program DSKINT. See Chapter 6 for a description of DSKINT.

4.2 PRIVILEGE

Certain accounts in the RSTS-11 system have special capabilities. These capabilities are outlined in this section. The accounts that have the special capabilities are called privileged accounts. An account is recognized as privileged by the RSTS-11 system if its project number of the project-programmer number group is a 1. The system library account [1,2], discussed in a preceding section of this chapter, is an example of a privileged account.

The system manager, operating under his own privileged account, designates accounts as privileged by assigning a project number 1 to the account when he creates it using the REACT system program. The available privileged account numbers are [1,3] through [1,254]. The assigned privileged accounts have the same special capabilities that the library privileged account [1,2] has. The user of an assigned privileged account also can create new accounts, as the system manager can. Privileged accounts, the system manager's as well as those he assigns, have the following special capabilities.

1. Unlimited file access. See Section 4.2.1.
2. Creation and modification of privileged compiled programs (with BAC filename extensions). See Section 4.2.2.
3. Use of privileged aspects of system programs. See Section 4.2.3.

4. Use of privileged SYS system functions and the PEEK function. See Section 4.3.

The above listed capabilities are described in the remaining sections of this chapter and in Chapter 5. It must be emphasized that there is no fail safe and that no error messages are generated if the use of such special capabilities are to result in destruction of the system. For this reason, it is suggested that the system manager assign privileged accounts sparingly. It is also recommended that the system manager create additional non-privileged accounts for himself and perform most of his functions under them. The system manager should use a privileged account only when necessary.

4.2.1 Unlimited File Access

No file in the RSTS-11 system can be protected against the user of a privileged account. A privileged user can create and delete files under any account number and can access files on LOCKED disks. Under such circumstances, no protection violation occurs. (Protection violation is a normal user recoverable error, number 10, as described in Appendix C of the BASIC-PLUS Language Manual.)

4.2.2 Creation and Modification of Privileged Programs

A program is considered privileged when it has a BAC filename extension and a protection code of <128> or greater. For example, during the building of the system library in the system generation procedures, the system manager designates the required system programs SYSTAT and TTYSET privileged by assigning to their compiled forms a protection code of <128> or greater. To effect the assignment of privilege, the system manager uses the NAME-AS statement. (Refer to Chapter 2 for the examples of designating SYSTAT and TTYSET privileged.) Any user of a privileged account can also designate a compiled program privileged.

If a program is designated privileged, any user can run the program in privileged program status (provided he has READ access to the file). The privileged program status exists for the duration of the program run. Privileged program status means that system operations, normally reserved to a user of a privileged account, are executed while running under a non-privileged account.

If the user running a privileged program interrupts execution of the program, such as by typing CTRL/C, the program loses its privileged status.

The ability to designate a program as privileged allows the system manager to control use of privileged functions as the system programs do. For example, the program TTYSET allows a user to change characteristics of his terminal. Such an action is a privileged system function executable only by owners of privileged accounts. With the privileged program status, execution of the function by the owner of a non-privileged account does not cause the normal program trap.

The same TTYSET program additionally allows a privileged user to change characteristics of a terminal other than his own. A check is built into the program to ensure that a user attempting to change the characteristics of a terminal other than his own is indeed a privileged user. As a result, the execution of privileged functions is made available to the non-privileged user but privileged features are restricted. The system manager likewise can control the use of privileged operations.

A privileged user can modify privileged BAC programs. The user of a non-privileged account never has write access to a privileged BAC program.

4.2.3 Use of Privileged Features of System Programs

The owner of a privileged account can execute the privileged features of system programs. Since the list of privileged features is lengthy, an entire chapter is devoted to explaining them. Certain programs, such as TTYSET, SYSTAT, and MONEY, are privileged but contain features helpful to the general user. These programs are therefore described in the RSTS-11 System User's Guide for the non-privileged user.

Two of the programs, TTYSET and MONEY, have privileged features about which the non-privileged user is not informed in the User's Guide. In Chapter 5 of this guide, only the privileged features of these programs are presented. The other program, SYSTAT, is discussed in Chapter 5 of this guide in greater depth than is done in the User's Guide although it contains no explicit privileged features. Thereby, the system manager can appreciate more fully the information returned by SYSTAT and be in a position to make sound judgments concerning the system.

4.3 SYSTEM FUNCTION CALLS AND PEEK

System function calls are separated into two classes, privileged and non-privileged. In the descriptions of each function call, the privileged ones are so labeled. The privileged system function calls can be used only by a privileged user, as defined in Section 4.2, or by a privileged program, as defined in Section 4.2.2. The non-privileged system function calls may be used by anyone and are completely safe in the sense that their use can do no damage to anyone other than the person who misuses them. In most cases, non-privileged system functions do no damage at all.

4.3.1 System Function Formats and Codes

The format of the call is:

```
V$=SYS(CHR$(F)+O$)
```

where:

V$ is the target string variable (See individual descriptions)
F is the SYS function code (See codes below)
O$ is the optional (by function) parameter string passed

CODE	FUNCTION EXECUTED BY THE SYS CALL
0	CANCEL ↑O EFFECT ON CONSOLE TERMINAL No argument string is needed. Target string = passed string This function call cancels the effect of a CTRL/O typed at the terminal. In addition, it is used with the programmable CTRL/C trap, FIP code-7. After a CTRL/C trap has occurred, teleprinter output is disabled until a SYS(CHR$(Ø)) is executed.
1	ENTER TAPE MODE ON CONSOLE TERMINAL No argument string is needed. Target string = passed string

This function is identical to typing the TAPE command on the terminal, as described in the RSTS-11 System User's Guide.

2 ENABLE ECHOING ON CONSOLE TERMINAL
(Also disable tape mode if present)
No argument string is needed.
Target string = passed string

Cancels the effect of function codes 1 and 3.

3 DISABLE ECHOING ON CONSOLE TERMINAL
No argument string is needed.
Target string = passed string

Prevents information typed at the terminal from being echoed automatically by the Monitor. Information typed, such as a password, is kept secret but accepted by the system as valid input.

4 ENABLE SINGLE CHARACTER INPUT MODE (ODT SUBMODE)
No argument string is needed.
Target string = passed string

Allows a single character to be accepted as input from the terminal. Normally, the system waits until a line delimited by a RETURN, LINE FEED, or ESCAPE has been typed before accepting input. In single character mode, a character typed at the terminal is passed immediately to the program by the next keyboard input request statement without waiting for the delimiter.

This function must be enabled prior to every input request statement that is to pass a single character to the program. A GET statement is used as the input request statement. (INPUT or INPUT LINE statements cause repeated generation of the input request until a line terminator is detected and must not be used.)

It is possible to use a SLEEP statement or a compute loop for a time delay between the SYS function call and the GET statement. Such a delay allows the operator time to type in more than one character before the system executes the GET statement.

Since this function is used in the system program ODT.BAS, it is sometimes referred to as "ODT submode".

5 EXIT TO EDITOR WITH NO "READY" MESSAGE
No argument string is needed.
No string is passed back.

Causes the same effect as the END statement, except that the READY message is not printed.

6 FIP function call (See Section 4.3.2.)

7 GET CORE COMMON STRING
No argument string is needed.
Target string = core common string

Allows a program to extract a single string from a data area loaded by another program running under the same job. The data area is called the core common and is from 0 to 127 8-bit bytes long. Refer to SYS function code 8.

8 PUT CORE COMMON STRING
Argument string is string for core common.
Target string = passed string

Allows a program to load a single string in a common data area called core common. This string can be extracted later by another program, running under the same job and called via the CHAIN statement. The string is from 0 to 127 8-bit bytes long. Refer to SYS function code 7.

This function provides a means for passing a limited amount of information when a CHAIN statement is executed. If a larger amount of information is to be passed, it must be written to a disk file and read back by the later program.

9 EXIT TO EDITOR AND SETUP "NONAME" PROGRAM
No argument string is needed.
No string is passed back.

This function causes the same actions as the END statement placed in the code, and, in addition, clears the program out of memory. This is the proper method of stopping a program that is not to be rerun. Also, the same action is performed by the command NEW NONAME.

4.3.2 SYS System Function Calls to FIP (Function Code 6)

The SYS system function call whose code is 6 is a more specialized case of the general system function call. It is specialized by an additional code, called the FIP code. The FIP code causes a call to be made to special non-resident code that performs File Processing.

The format of the call is:

```
V$=SYS(CHR$(6%)+CHR$(F0)+O$)
```

where:

```
V$  is the target string variable (See below.)
F0  is FIP function (See codes below.)
O$  is the optional (by function) parameter string.
```

The general format of the target variable (V$) is:

BYTE 1	JOB NUMBER TIMES 2
BYTE 2	VALUE OF INTERNAL FUNCTION CALLED (MEANINGLESS TO GENERAL USER)
BYTES 3-30	BYTES COPIED FROM BYTES 4-37 (OCTAL) OF FIRQB (SEE INDIVIDUAL DESCRIPTIONS)

NOTE

30 bytes are always passed back.

The proper use of the FIP system function call requires that the user program build a parameter string to pass and that the program later extract the data from the returned string, called the target string. Each call returns a string of 30 bytes, each byte (or character) of which may or may not contain useful information. The descriptions of the FIP codes specify the contents of each byte in the string, from which the user determines whether the information contained is of interest.

A recommended method of building a 30-byte string to pass is to dimension a 30-byte list and set the items in the list to values which map into those required in the optional parameter string format. The list can later be set to a character string by the CHANGE statement before it is passed as the parameter string of the FIP system function call. The resulting character string is in proper format and contains the correct byte values so that it may be placed as the optional parameter string of the FIP system function call. For example:

```
10 DIM A%(30)          !the string is 30 bytes long
20 A%(0)=30            !0th element is length of list
30 A%(1)=6             !SYS call code 6 (FIP call)
40 A%(2)=6             !FIP code 6 (attach)
50 A%(3)=10            !Job number times 2 of job to
                       !attach to (Job 5)
                       !
   :
```

The remainder of the bytes are supplied according to the requirements of FIP call code 6 and the user's requirements.

Following the code which builds the list is the CHANGE statement and the call itself, as shown below.

```
CHANGE A% TO A$          !generates character
      .                  !string from the integer
      :                  !list
B$ = SYS(A$)             !invoke system function
                         !call
```

In the case of the FIP call using FIP code 6, the action performed (attach) is of importance and the string returned does not contain information that is worthwhile to the user.

Other FIP system function calls return a target string which is of prime interest to the user program. In such a case, the user program transforms the returned byte string to the list of integer items. For example, the statement

```
CHANGE SYS(A$) TO A%
```

performs the conversion so that the data of the individual items can be extracted and converted to meaningful form.

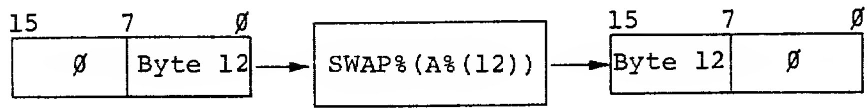
In the instance of a FIP code of 15, as an example, bytes 11 and 12 of the returned string are the file name extension encoded in Radix-5Ø format. The integer representation of each byte, however, occupies a full word, 16 bits in length. Thus, list items number 11 and 12 would appear as the following:

	15 7	7 Ø
A%(11)	Ø	Byte 11
	15 7	7 Ø
A%(12)	Ø	Byte 12

Byte 11 contains the low byte portion of the Radix-50 word; byte 12 contains the high byte portion of the Radix-5Ø word. The two bytes must be combined into a single word and converted to the proper character string representation. This is accomplished by the following:

S$ = RAD$(A%(11) + SWAP%(A%(12)))

The SWAP% function reverses the bytes (the low byte takes the high byte position and vice-versa) in an integer word. Graphically, the operation appears as follows:



Thus, byte 12 takes the high byte position in the word. The two words are then combined by the + operator to form one word. The RAD$ function performs the conversion on that one integer word to produce the 3-character string representation of the file name extension. The character string is assigned to the character variable S$. S$ is converted to a usable format.

The remainder of the section describes the individual FIP system function calls and the structures of the parameter string options and of the returned target string.

DESCRIPTIONS BY FUNCTION CODE (FØ)

-15 ERROR LOGGING

Reserved for DIGITAL use.

-14 SYSTEM DATE AND TIME CHANGER

O$=CHR$(HDA)+CHR$(LDA)+CHR$(HTI)+CHR$(LTI)

where:

L means Low byte and H means High byte
DA means system date
TI means time of day

No useful data is returned.

This function changes the monitor date and time of day values which are returned by the DATE$(Ø%) and TIME$(Ø%) functions in BASIC-PLUS.

-13 PRIORITY AND QUANTUM FACTOR CHANGER

O$=CHR$(J%)+CHR$(P%)+CHR$(QM)+CHR$(QA)

where:

J% is the job number. Ø means the current job.

P% is the priority value between -128 (lowest) and +127 (highest)

QM is the quantum multiplier factor.

QA is the quantum adder factor.

No useful data is returned.

The user program can supply values to alter the job's chance of running and to increase or decrease the amount of time the system allows the job to run. The system automatically sets a job's priority to -1. (During the LOGIN procedure, a job's priority is temporarily Ø.) The system allows a job to run for N number of clock interrupts (the time quantum) according to the following algorithm.

$$N = ((J) * 2^{QM}) + QA$$

where:

J	=	job size in K words.
QM	=	quantum multiplier factor which can be between Ø and 127. If QM=Ø, then N=QA.
QA	=	quantum adder factor which must be between Ø and 127.

The system automatically sets QM to 1 and QA to 4 for all jobs.

-12 Not used.

-11 BACKUP FILE STAT CHANGER (PRIVILEGED)

O$=CHR$(N)+CHR$(LDLA)+CHR$(HDLA)+CHR$(LDC)+CHR$(HDC)+
CHR$(LTC)+CHR$(HTC)

N=CHANNEL NUMBER (1 TO 12)
L PREFIX IS LOW BYTE: H PREFIX IS HIGH BYTE
DLA=DATE OF LAST ACCESS
DC=DATE OF CREATION
TC=TIME OF CREATION

No useful data is returned.

This function changes the date of last access, date of creation, and time of creation fields for a disk file.

-10 FILE NAME STRING SCAN

O$=STRING TO SCAN

Loaded FIRQB is passed back as follows:

BYTE	1	JOB NUMBER TIMES 2
BYTES	2-4	?
BYTE	5	PROG NUMBER OR Ø
BYTE	6	PROJ NUMBER OR Ø
BYTES	7-1Ø	FILE NAME OR Ø (RADIX-5Ø FORMAT)
BYTES	11-12	EXTENSION OR Ø (RADIX-5Ø FORMAT)
BYTES	13-2Ø	Ø
BYTE	21	Ø IF NO PROTECTION; ELSE 255
BYTE	22	PROTECTION IF SPECIFIED
BYTES	23-24	DEVICE NAME OR Ø
BYTE	25	DEVICE NUMBER IF SPECIFIED
BYTE	26	Ø IF NO NUMBER; ELSE NON-Ø
BYTES	27-3Ø	Ø

This function causes the string passed to be scanned and converted into the form necessary for further FIP system function calls.

-9 HANGUP A DATASET (PRIVILEGED)
O$=CHR$(N)+CHR$(?)
N=KEYBOARD NUMBER OF LINE TO HANGUP (MUST BE DC11 LINE)
?=Anything such that an even number of bytes is passed.
No useful data is returned.

-8 GET OPEN CHANNEL STATISTICS
O$=CHR$(N)
N=CHANNEL NUMBER (0 TO 15)
Data returned:

BYTES 3-4	WORD 1 OF FCB/DDB
BYTES 5-6	WORD 2 OF FCB/DDB
...	
BYTES 23-24	WORD 11 OF FCB/DDB
BYTES 25-26	WORD 12 OF FCB/DDB
BYTES 27-28	WORD 16 OF FCB/DDB
BYTES 29-30	WORD 15 OF FCB/DDB

-7 CONTROL/C TRAP ENABLE
No parameters are needed.
Nothing worthwhile is passed back.

After this FIP function is executed in a user program, the Run Time System treats the first CTRL/C subsequently typed on any terminal belonging to the job as a trappable error (ERR=28). Upon execution of the trap, the terminal is placed in CTRL/O mode (all output to the terminal is suppressed), and control is immediately passed to the numbered program statement which has been designated as the error-handling routine by the last execution of an ON ERROR GOTO statement. After the trap, the user program must issue a SYS call (code 0) to re-enable programmed output at the terminal.

Such trapping of CTRL/C, however, guarantees only that a defined set of statements are executed when CTRL/C is typed. It is not possible to resume execution at the exact point where the CTRL/C occurred.

If certain critical sections of BASIC-PLUS code are to be completely immune to possible CTRL/C aborts, three actions must occur.

(a) The job must detach itself from its terminal. See the description of FIP code +7.

(b) The program must have CLOSEd all channels on which other terminals in the job had been OPENed.

(c) The job must have DEASSIGNed any terminal which had been previously ASSIGNed to it. See the description of FIP code +11.

If the three actions occur, program execution under the job proceeds immune to any CTRL/C.

After the job has completed its critical processing in the detached state, one of two actions can occur.

(a) The job can kill itself by means of FIP code +8.

(b) The job can find a free terminal (presumably the one from which it detached itself), and "force" into that terminal input buffer the character strings needed for logging into the system and attaching the job to the terminal. (See the descriptions of FIP codes -4, +4, and +6.)

-6 POKE CORE (PRIVILEGED)
O$=CHR$(LA)+CHR$(HA)+CHR$(DATA)+...
LA=low byte of address
HA=high byte of address
Data is the data byte stream.
No useful data is returned.

This FIP function changes the contents of memory and can destroy the system very easily. It is the most dangerous function and must be used with care.

-5 BROADCAST TO TERMINAL (PRIVILEGED)
O$=CHR$(N)+M$
N=Keyboard number to receive output message
M$=Message to broadcast (must not be null string)
No useful data is returned.

-4 FORCE INPUT TO TERMINAL (PRIVILEGED)
O$=CHR$(N)+I$
N=Keyboard number to receive forced input
I$=String to force to terminal (must not be null string)
No useful data is returned.

-3 GET MONITOR TABLE LOCATIONS
No parameters are needed.
Data returned:

BYTE	3	MAXIMUM KEYBOARD NUMBER ON SYSTEM
BYTE	4	MAXIMUM JOB NUMBER ON SYSTEM
BYTES	5-6	"DEVCNT"
BYTES	7-8	"DEVPTR"
BYTES	9-10	"CORTBL"
BYTES	11-12	"JOBTBL"
BYTES	13-14	"JBSTAT"
BYTES	15-16	"JBWAIT"
BYTES	17-18	"UNTCLU"
BYTES	19-20	"UNTCNT"
BYTES	21-22	"SATCNT"
BYTES	23-24	"NAME1"
BYTES	25-26	"NAME2"
BYTES	27-28	TODAY'S DATE IN INTERNAL FORM

NOTE

The following (decimal) locations are fixed in the resident monitor area in core for version 4A. They may be examined with the PEEK function.

```
IDATE  36
ITIME  38
DATE   2048
TIME   2050
JOB    2056 (BYTE)
JOBDA  2098
FREES  2108
       2110
FREEB  2112
       2114
DEVNAM 2206
```

-2 DISABLE FURTHER LOGINS (PRIVILEGED)
No parameters are needed.
No useful data is returned.

-1 ENABLE FURTHER LOGINS (PRIVILEGED)
No parameters are needed.
No useful data is returned.

NOTE

Function codes 0 on up require filling called for bytes with 0's if parameter string (O$) is too short.

0 CREATE USER ACCOUNT (PRIVILEGED)
Data passed:

```
BYTE    7       PROGRAMMER NUMBER (0 TO 255)
BYTE    8       PROJECT NUMBER (1 TO 255)
BYTES   9-12    PASSWORD IN RADIX-50 (ONLY IF SYSTEM ENTRY)
BYTES   13-14   DISK QUOTA IN 256 WORD BLOCKS (0 MEANS NO QUOTA)
BYTES   23-24   DISK TYPE (DF,DK,DP - 0 MEANS SYSTEM)
BYTE    25      DISK UNIT (0 TO N) -- IF BYTES 23-24 ARE 0
                                      THEN THIS IS 0
BYTE    26      255                -- THIS BYTE IS 255 IF (AND
                                      ONLY IF) BYTE 25 CONTAINS
                                      A REAL UNIT DESIGNATION.
                                      IN THIS WAY WE DISTIN-
                                      GUISH BETWEEN 'DK:' AND
                                      'DK0:'.

BYTES   27-28   UFD CLUSTER SIZE (1 TO 16 - 0 DEFAULTS)
```

No useful data is returned.

1 DELETE USER ACCOUNT (PRIVILEGED)
Data passed:

```
BYTE    7       PROGRAMMER NUMBER -- YOU CANNOT DELETE
BYTE    8       PROJECT NUMBER    -- [1,1],[1,2],[0,1]
BYTES   23-24   DISK TYPE (DF,DK,DP - MEANS SYSTEM)
```

No useful data is returned.

2 CLEANUP A DISK PACK (PRIVILEGED)
Data passed:

```
BYTES   23-24   DISK TYPE (DF,DK,DP - 0 IS SYSTEM)
BYTE    25      DISK UNIT (0 TO N)  - IF BYTES 23-24 ARE 0
                                      THEN THIS IS 0
```

```
        BYTE   26       255         -- THIS BYTE IS 255 IF (AND ONLY
                                       IF) BYTE 25 CONTAINS A REAL
                                       UNIT DESIGNATION.  IN THIS WAY
                                       WE DISTINGUISH BETWEEN 'DK:'
                                       AND 'DK0:'.
        No useful data is returned.

3       DISK PACKS AND TERMINAL STATUS (PRIVILEGED)
        Data passed:
        BYTE   3        ODD =:: TERMINAL STATUS
                        0   =:: MOUNT DISK PACK
                        2   =:: DISMOUNT DISK PACK
                        4   =:: LOCK OUT DISK PACK
                        6   =:: UNLOCK DISK PACK

        For terminal status (0 means no change):
        BYTE   4        255=::YOUR TERMINAL ; N=::KBN
        BYTE   5        NEW FORM WIDTH (1 TO 255)
        BYTE   6        128=::TAB ; 255=::NO TAB
        BYTE   7        128=::NO FORM ; 255=::FORM
        BYTE   8        128=::LC ; 255=::NO LC
        BYTE   9        128=::NO XON ; 255=::XON
        BYTE   10       128=::ECHO ; 255=::NO ECHO
        BYTE   11       128=::NO SCOPE ; 255=::SCOPE
        BYTE   12       128=::NO 175,176 ; 255=::175,176
        BYTE   13       255=::SERIAL LA30 FILL
                        1 TO 7=::0 TO 6 FILL FACTOR
        BYTE   14       1 TO 4 FOR DC11 SPEEDS 0 TO 3
        No useful data is returned.

        For MOUNT, DISMOUNT,LOCK,UNLOCK:
        BYTES  23-24    DISK TYPE (DF,DK,DP)
        BYTE   25       DISK UNIT (0 TO N)
        BYTE   26       255

        For MOUNT:
        BYTES  7-10     PACK ID IN RADIX 50

        No useful data is returned.

4       LOGIN (PRIVILEGED)
        Data passed:
        BYTE   7        PROGRAMMER NUMBER -- NEVER CAN
        BYTE   8        PROJECT NUMBER    -- BE [0,1]
        BYTES  9-12     PASSWORD IN RADIX 50
        Data returned:
        BYTE   3        NUMBER OF JOBS THIS ACCOUNT (>=1)
        BYTES  4-N      JOB NUMBERS (*2) OF DETACHED JOBS (0 IS END)

5       LOGOUT (PRIVILEGED)
        No argument string is needed.
        No useful data is returned.

6       ATTACH (PRIVILEGED)
        Data passed:
        BYTE   3        JOB NUMBER TIMES 2 TO ATTACH TO
        BYTES  7-8      PPN OF JOB TO ATTACH TO
        BYTES  9-12     PASSWORD OF PPN IN RADIX 50
        No useful data is returned.
```

```
7         DETACH (PRIVILEGED)
          No argument string is needed.
          No useful data is returned.

8         CHANGE PASSWORD/QUOTA OR KILL JOB (PRIVILEGED)
          Data passed for change:
          BYTES  7-8      PPN OF ACCOUNT TO CHANGE (0 MEANS YOU)
          BYTES  9-12     NEW PASSWORD IN RADIX 50 (0 MEANS SAME)
          BYTES  13-14    NEW DISK QUOTA IN BLOCKS
          BYTE   21       255 IF QUOTA IS REAL
          BYTE   28       0

          Data passed for KILL:
          BYTE   3        JOB NUMBER
          BYTE   28       255
          No useful data is returned.

9         ERROR MESSAGES
          Data passed:
          BYTE   3        ERROR NUMBER (0 TO 127)
          Data returned:
          BYTE   2        KEYBOARD NUMBER TIMES 2 ** EXCEPTION **
          BYTES  3-30     MESSAGE ENDING WITH A NULL (0) BYTE

10        ASSIGN A DEVICE
          Data passed:
          BYTES  23-24    DEVICE NAME (DT,PR,PP,MT,CR,LP,KB)
          BYTE   25       DEVICE UNIT (0 TO N) -- BYTE 25 IS USED IF
          BYTE   26                            -- BYTE 26 IS 255
          No useful data is returned.

11        DEASSIGN A DEVICE
          See ASSIGN A DEVICE above.

12        DEASSIGN ALL DEVICES
          No argument string is needed.
          No useful data is returned.

13        ZERO A DEVICE
          Data passed:
          BYTES  5,6      PPN IF PRIVILEGED (0 IS YOUR'S)
          BYTES  23-24    DEVICE NAME (DT,DF,DK,DP - 0 MEANS SYSTEM)
          BYTES  25       DEVICE UNIT (0 TO N) -- IF BYTES 23-24 ARE 0
                                                  THEN THIS IS 0
          BYTE   26       255                  -- THIS BYTE IS 255 IF
                                                  (AND ONLY IF) BYTE 25
                                                  CONTAINS A REAL UNIT
                                                  DESIGNATION.  IN THIS
                                                  WAY WE DISTINGUISH
                                                  BETWEEN 'DK:' AND
                                                  'DK0:'.
          No useful data is returned.

14        READ/RESET ACCOUNTING DATA
          Data passed:
          BYTES  3-4      INDEX (1 TO N) OR 0 FOR LOOKUP ON PPN
          BYTES  5-6      0 FOR READ ONLY/NON-0 FOR READ AND RESET
          BYTES  7-8      PPN IF LOOKUP ELSE ANYTHING
          BYTES  23-24    DISK TYPE (DF,DK,DP - 0 MEANS SYSTEM)
          BYTE   25       DISK UNIT (0 TO N)  - IF BYTES 23-24 ARE 0
                                                THEN THIS IS 0
```

```
BYTE   26       255           -- THIS BYTE IS 255 IF (AND ONLY
                                 IF) BYTE 25 CONTAINS A REAL
                                 UNIT DESIGNATION.  IN THIS WAY
                                 WE DISTINGUISH BETWEEN 'DK:'
                                 and 'DKØ:'.
Data returned:
BYTES  5-6      DISK OWNED BY ACCOUNT
BYTES  7-8      PPN OF ACCOUNT
BYTES  9-12     PASSWORD IN RADIX 50 OF ACCOUNT
BYTES  13-14    CPU TIME (TENTHS OF SECONDS)
BYTES  15-16    CONNECT TIME (MINUTES)
BYTES  17-18    KILO-CORE-TICKS (1K'S*TENTHS OF SECONDS)
BYTES  19-20    DEVICE TIME (MINUTES)
BYTE   21       NUMBER OF LOGINS
BYTE   22       NUMBER OF LOGOUTS
BYTES  27-28    DISK QUOTA IN BLOCKS (0 MEANS NO QUOTA)
BYTES  29-30    UFD CLUSTER SIZE
If not privileged then forces:
BYTES  3-4      TO 0 (DO LOOKUP)
BYTES  5-6      TO 0 (READ ONLY)
BYTES  7-8      TO CURRENT PPN FOR LOOKUP
```

```
15        DIRECTORY INFORMATION
          Data passed:
          BYTES  3-4      INDEX (0 TO N)
          BYTES  5-6      PPN (0 MEANS CURRENT PPN)
          BYTES  23-24    DEVICE TYPE (DT,DF,DK,DP - 0 MEANS SYSTEM)
          BYTE   25       DEVICE UNIT (0 TO N)   -- IF BYTES 23-24 ARE 0
                                                    THEN THIS IS 0
          BYTE   26       255                    -- THIS BYTE IS 255 IF
                                                    (AND ONLY IF) BYTE 25
                                                    CONTAINS A REAL UNIT
                                                    DESIGNATION.  IN THIS
                                                    WAY WE DISTINGUISH
                                                    BETWEEN 'DK:' AND
                                                    'DK0:'.

          Data returned:
          BYTES  7-10     FILE NAME IN RADIX 50
          BYTES  11-12    EXTENSION IN RADIX 50
          BYTES  13-14    LENGTH IN BLOCKS/SECTORS
          BYTE   15       PROTECTION
          BYTES  17-18    DISK - DATE OF LAST ACCESS
                          TAPE - DATE OF CREATION
          BYTES  19-20    DISK - DATE OF CREATION
          BYTES  21-22    DISK - TIME OF CREATION
          BYTES  27-28    DISK - FILE CLUSTER SIZE
          BYTES  29       NUMBER OF ENTRIES (DISK=8; TAPE=6)
```

4.3.3 The PEEK Function

There is one final privileged function in RSTS-11. This is the PEEK function, which permits the user to examine words in the PDP-11's core memory. It is called as follows:

```
I%=PEEK(J%)
```

PEEK takes an integer argument and returns an integer value, which is simply the contents of the core location specified by the integer argument. Since on the PDP-11 words are addressed always as even numbers (odd addresses are byte addresses, not word addresses), the user must be careful to always specify an even integer argument to PEEK.

PEEK is privileged since any attempt to access an odd address (i.e., failing to use an even integer argument) or attempting to access nonexistent memory will cause a catastrophic error.

The PEEK function is normally used in conjunction with the system function call that returns various monitor table locations.

CHAPTER 5

PRIVILEGED FEATURES OF SYSTEM PROGRAMS

A system program is coded in BASIC-PLUS, and stored either as a disk-resident system library file in compiled form under account [1,2] or as a file in source form in the RSTS-11 kit. System library files are described in Section 2.4.4 of the RSTS-11 System User's Guide, and the system library account [1,2] is described in Section 4.1.2 of this guide. The system manager determines the contents of the system library during the system generation procedures by including in or omitting from the installation's system library account those system programs supplied by DIGITAL.

This chapter covers those features of the system programs which the system manager or privileged user needs to be aware of to operate and manage the RSTS-11 system. The description of the system programs given in Chapter 4 of the RSTS-11 System User's Guide is presented in a modular format such that the system manager, as he deems necessary, can remove a description of a system program from that guide without affecting the descriptions of other system programs. As a result, the system manager can determine what the user at the local installation needs to know about the individual system programs provided on the RSTS-11 system.

Other system programs are documented in formatted ASCII files with DOC filename extensions in the RSTS-11 kit. These so called documentation files can be transferred from the RSTS-11 kit to a keyboard or line printer by using the PIP system program and its /FA switch. See Chapter 4 of the RSTS-11 System User's Guide for a description of PIP.

Many of the system programs make use of SYS system function calls. Therefore, a firm understanding of the techniques and strategy in the use of system programs leads to an easier understanding of the operation of certain SYS system function calls.

References are made in the discussion of SYS system function calls to programs described in this chapter as an aid in grasping the concept of coding SYS calls.

The format of the program descriptions in this chapter is mainly expository. Reference aids have been included in appending the relevant system program filename to the section heading and by inclusively tabulating all relevant program commands or options. The description of each program informs the system manager how the related system program must be stored and with what protection code.

5.1 CONTROLLING SYSTEM START UP - INIT

The user can control system start up by means of the system initialization program INIT. System start up occurs when the user executes the START option or when the monitor attempts an automatic restart after a system crash. At start up time, the monitor automatically runs INIT which, in turn, executes special commands on the system. By the ability to specify these special commands, the user controls the actions which occur at system start up.

To control system start up efficiently, the system manager must understand the conditions in effect at start up time. The following conditions pertain.

a) Login attempts are prohibited (the monitor disables the login capability).

b) The monitor logically mounts only the system disk.

c) No output is made to any terminal.

d) The monitor logs the console terminal (KBØ:) onto the system under the system library account [1,2].

e) At the console terminal, the monitor executes the command equivalent to CHAIN "INIT" or CHAIN "INIT" 1ØØ.

As a consequence of condition (e), INIT runs and reads one of two control files on the system disk (SYØ:) under account [1,2]: either START.CTL or CRASH.CTL. The two control files contain special commands for the initialization of the system for time sharing. The following sections describe the INIT commands and their usage.

5.1.1 INIT Program Commands

The RSTS system is not fully initialized until INIT runs and executes commands in the control file. By specifying the INIT commands described in Table 5-1, the system manager can control system start up according to requirements at the local installation. For example, the system possibly uses other disk devices in addition to the system disk. By means of the MOUNT command, the system manager can make such devices in the public and in the private structure immediately available to users before they can log into the system. The local start up procedures must include making the specified devices physically ready on the proper drive units.

To execute other actions on the system, the system manager can cause INIT to execute BASIC-PLUS commands and programs. Executing the LOGINS command enables further logins on the system. The LOGIN command automatically logs a specified job onto the system at a designated terminal to allow execution of commands and programs. For example, the FORCE command can run the TTYSET system program at the terminal and can subsequently execute commands to establish terminal characteristics of certain keyboards. The SEND command prints text on all on-line terminals.

The following sample START.CTL file and accompanying explanation shows the usage of INIT commands.

```
MOUNT     DK1:PACK1                  (line a)
MOUNT     DK2:PRIV1                  (line b)
LOGINS                               (line c)
LOGIN     KB1:  [1,5]                (line d)
FORCE     KB1: RUN $TTYSET           (line e)
FORCE     KB1: KB1:                  (line f)
FORCE     KB1: VTØ5                  (line g)
FORCE     KB1: EXIT                  (line h)
FORCE     KB1: BYE F                 (line i)
FORCE     KBØ: RUN $ERRCPY           (line j)
SEND      RSTS IS NOW ON THE AIR ... (line k)
END                                  (line l)
```

Table 5-1

Control File Contents

Command Name and Format[1]	Use
LOGINS	Allows users at both local and remote terminals to type requests to the LOGIN system program.
SEND␣xxx	Transmits the optional text xxx to all keyboards currently online except the console keyboard (KBØ:).
LOGIN␣KBn:␣[n,m]	Logs the terminal specified by KBn: onto the system using the account indicated by [n,m]. INIT automatically looks up the password. NOTE Cannot be used on KBØ: because INIT must run on that terminal.
FORCE␣KBn:␣xxx	Causes the text xxx to be placed in the input buffer of the terminal specified by KBn: and executed as if typed at that terminal. The text can be any BASIC-PLUS command or TTYSET system program SET commands. The LOGINS command must appear before the FORCE commands can be executed. NOTE If the ↑ character is the first character of the text xxx, a CTRL/C is placed in the terminal buffer ahead of the specified text xxx. However, INIT generates an error if an attempt is made to force a ↑C to KBØ.
MOUNT␣dev:id	Causes the disk unit specified by the device designator dev: and by the pack identification (id) to be logically recognized by the RSTS system. Additionally, the MOUNT command as used in the control file causes a clean operation (if necessary) and an unlock operation. Refer to Section 5.3.2 for information concerning mount, clean and unlock operations.
BYE or END	Causes execution of the INIT system program to be terminated, BYE causes the job running under account [1,2] to be logged out, thus freeing the console keyboard for other use and preventing unauthorized use of account [1,2]. END must be used in place of BYE when running ERRCPY since END does not logout the job running ERRCPY under account [1,2]. See Section 5.1Ø for a description of ERRCPY.

[1]The notation ␣ indicates that a space character is required.

The user makes other disks available on the system as shown at lines a and b. Line a causes the system to recognize the additional disk cartridge PACK1 of the public structure on RK11 drive unit 1. The user must mount all public disks (except the system disk) in this manner if all user files are to be available immediately. The system also cleans (if necessary) and unlocks PACK1 so that users can create files on it.

Line b causes the same results as line a but for the disk cartridge PRIV1 in the private structure.

Logins must be enabled before attempting to log a terminal onto the system. The LOGINS command at line c is required to enable logins and to allow users access to the system.

The command at line d logs in a job at keyboard number 1 under account [1,5]. INIT automatically looks up the password of the account. In this manner, secrecy is maintained by not requiring passwords in control files.

The commands at lines e through h run the TTYSET system program to set terminal characteristics. For more information on TTYSET, see Section 5.7.

At line i, keyboard number 1 is logged off the system to prevent unauthorized use of the account.

The command at line j causes the console terminal (KBØ:) to run the ERRCPY system program under account [1,2]. This action enables the RSTS system to take advantage of hardware error logging on the system.

The SEND command notifies users that time sharing operations have begun. The message is printed on all terminals on line to RSTS.

The END command at line 1 terminates the INIT program running at the console terminal. However, the command leaves KBØ: logged into the system so that the system can execute the command forced into the keyboard buffer by the command at line j. Since ERRCPY detaches from the terminal, KBØ: does not remain logged into the system after initialization.

5.1.2 Creation and Usage of Control Files

The INIT system program control files START.CTL and CRASH.CTL must contain commands to properly initialize the RSTS system and must be stored on the system disk (unit Ø of the public structure). The files included in the RSTS system generation kit are only samples and, without modification, may not execute properly on a given system. The system manager must ensure that the files include the necessary commands to initialize the local installation. For example, to replace the sample START.CTL file supplied with the RSTS kit, run PIP and proceed as follows.

```
RUN $PIP
PIP RSTS VØ4B-17 SYSTEM #88Ø
#SYØ:START.CTL$<KB:/FA
        (user types his new INIT commands.)
↑Z
#
```

The procedure shown ensures that the new file is created on unit Ø (SYØ:). This is important since, when INIT runs, only the system disk in the public structure is mounted.

It is important that both control files perform the following functions:

a) Mount all non-system public disks,
b) enable logins,
c) set keyboard characteristics for non-ASR33 terminals, and
d) run any system programs which must execute during time sharing.

Since, at start up time, the system sets the characteristics of all terminals to those of an ASR33-type terminal, the user must run the TTYSET system program to determine the correct characteristics. For example, if keyboards 1 and 3 are LA3ØS-type terminals to run at 3ØØ baud, if keyboard 2 is an ASR33-type terminal, and if keyboard 4 is a VTØ5 to run at 3ØØ baud; the system manager can specify the following sequence of commands to properly set the characteristics.

```
LOGIN KB4: [1,5]
FORCE KB4: RUN $TTYSET
FORCE KB4: KB4:
FORCE KB4: VTØ5
FORCE KB4: KB1:
FORCE KB4: LA3ØS
FORCE KB4: KB3
FORCE KB4: LA3ØS
FORCE KB4: EXIT
```

The sequence of INIT commands shows the optimum use of a keyboard on which to execute commands. Before forcing the next command to a terminal, INIT waits until the terminal output buffers are empty and the job is in the keyboard wait state. Therefore, it is advantageous to force commands to a terminal which generates output at the highest speed since that terminal's output buffers empty faster. Note, in the example, that keyboard 4 is established as a VTØ5 whose default output speed is 3ØØ baud. The interface speed is the important factor since it is not necessary that a terminal actually be connected to the interface unless a visual record of the start up procedure is desired.

In the case of INIT running as a result of a system crash, it is important that commands be executed to obtain crash dump information. To discover the cause of a system crash, the system manager can specify that INIT run the crash analysis program ANALYS. The following commands inserted in the CRASH.CTL file ensure that the analysis occurs.

```
LOGIN KB4: [1,5]
FORCE KB4: RUN $ANALYS
FORCE KB4: [Ø,1] CRASH.DMP
FORCE KB4:
```

For more information on ANALYS, see Section 5.8.

The system manager can specify commands to be executed on KBØ: even though INIT runs at the console terminal. The system assigns job number 1 to INIT since it is the first job to run at start up time. Because INIT runs at the console terminal, commands forced to KBØ: are not executed until INIT terminates. To prevent premature termination, INIT does not allow a CTRL/C combination to be forced to the console terminal. Therefore, the user must not specify the ↑ character in a FORCE command on the console terminal.

If the system manager wishes to run a certain job as job number 1, he can specify the proper commands to the console terminal. For example, it is often desired to run the ERRCPY system program as job 1. The following command in the control file ensures this action.

```
FORCE KBØ: RUN$ERRCPY
```

ERRCPY runs and detaches immediately after INIT terminates as the result of the END command.

5.1.2.1 START.CTL File Example - The following is an example of a START.CTL file which sets keyboards 1 and 2 and runs the ERRCPY program.

```
MOUNT DK1:PRIV1
MOUNT DK2:PRIV2
LOGINS
LOGIN KB1: [1,2]
FORCE KB1: RUN $TTYSET
FORCE KB1: KB1:
FORCE KB1: VTØ5
FORCE KB1: KB2:
FORCE KB1: LA3ØS
FORCE KB1: EXIT
FORCE KBØ: RUN $ERRCPY
SEND RSTS IS NOW ON THE AIR ...
END
```

5.1.2.2 CRASH.CTL File Example - The following is an example of a CRASH.CTL file which runs the appropriate programs to recover crash information and initializes the system for time sharing.

```
MOUNT DK1:PRIV1
MOUNT DK2:PRIV2
SEND RSTS RECOVERING FROM A CRASH...
LOGINS
LOGIN KB1: [1,2]
FORCE KB1: RUN $TTYSET
FORCE KB1: KB1:
FORCE KB1: VTØ5
FORCE KB1: KB2:
FORCE KB1: LA3ØS
FORCE KB1: EXIT
FORCE KB1: RUN $ANALYS
FORCE KB1:
FORCE KB1: ANALYS.TMP
FORCE KB1: RUN $ERRCRS
FORCE KB1: ERRCRS.TMP
FORCE KB1:
```

```
FORCE KB1: RUN $ERRDIS
FORCE KB1: ERRCRS.TMP
FORCE KB1: ERRDIS.TMP
FORCE KB1: ALL/S
FORCE KB1: ALL
FORCE KB1: /K
FORCE KB1: BYE
FORCE KB1: YES
FORCE KBØ: RUN $ERRCPY
SEND RSTS IS NOW ON THE AIR..
END
```

5.1.2.3 Simplified CRASH.CTL File Example - The following is an example of a CRASH.CTL file which demonstrates how INIT can run itself and therefore initialize the system for time sharing.

```
SEND RSTS RECOVERING FROM A CRASH...
LOGINS
FORCE KB1: SET VTØ5
LOGIN KB1: [1,2]
FORCE KB1: RUN $ANALYS
FORCE KB1:
FORCE KB1: ANALYS.TMP
FORCE KB1: RUN $ERRCRS
FORCE KB1: ERRCRS.TMP
FORCE KB1:
FORCE KB1: RUN $ERRDIS
FORCE KB1: ERRCRS.TMP
FORCE KB1: ERRDIS.TMP
FORCE KB1: ALL/S
FORCE KB1: ALL
FORCE KB1: /K
FORCE KBØ: RUN $INIT
END
```

5.2 PERFORMING SYSTEM SHUT-DOWN

The shut-down procedures for the RSTS-11 system are the most critically important after the system generation procedures. If system shut-down is not conducted in an orderly and careful fashion, much valuable user data can be irretrievably lost. It is, therefore, strongly suggested that the system manager enforce rigid adherence to the proper shut-down procedures.

To fully understand shut-down procedures, knowledge of other RSTS-11 system procedures is necessary. Before implementing the procedures described in this section, the system manager must familiarize himself with the concepts presented elsewhere in this chapter under the titles "On-Line System Control", "Disk Management", and "Monitoring System Status".

The following steps are mandatory for readying the system for orderly shut-down procedures. All further logins must be disabled by use of the NO LOGINS command of UTILTY. The system manager does not want new users entering the system when shut-down procedures are being initiated. All non-system disk devices must be locked to prevent files from being opened on them. The system manager does this by using the LOCK command of UTILTY. For the same purpose, the system manager must assign all other devices (by the ASSIGN system command) to his job. This action ensures that users already logged into the system do not initiate further processing on a device.

When all system resources are prohibited from further use, the system manager must next advise users that shut-down procedures are in effect. The SEND command of UTILTY is the means to accomplish transmitting the advice. The user should be informed that he has a certain length of time to close all his files, deassign all his devices and log off the system.

The system manager next monitors system status to determine what devices have been freed. When an assignable device is no longer assigned or initialized, he uses the ASSIGN command to make it unavailable to other users. If a disk device has no open files, the system manager can logically dismount the disk using the UTILTY system program command DISMOUNT. (In no case is the system disk dismounted, however.)

When all devices have been assigned to the system manager's job and all disks (except the system disk) have been logically dismounted, the system manager next ensures that no jobs are active except his own. He does so by requesting a job status report via the SYSTAT system program. If all jobs are logged out (except the system manager's), if all non-system disks are logically dismounted, and if further logins are disabled, the system manager can safely proceed to shut down the system.

The above conditions being true, the system manager shuts down the RSTS-11 system in the following way. He logs his job off the system. (The BYE command automatically deassigns all the devices assigned to a user's job number.) When the printing of LOGOUT messages is completed, and the Teletype is idle, the system shut-down is effected by moving the CPU ENABLE/HALT switch to its HALT position. Alternatively, he can use the SHUTUP system program which logs his job off the system and, additionally,brings the system to an orderly halt at address 54.

It can be seen, from the above discussion, that it is necessary to establish administrative procedures governing RSTS-11 system operational hours. All users can be kept informed by means of the NOTICE.TXT file which prints out when a user successfully logs into the system. System shut-down times are to be considered fixed so that users can plan a work load and properly complete their processing within the allotted hours of scheduled operational time. Erratic system shut-downs are guaranteed to cause unhappy users.

5.3 SYSTEM UTILITY OPERATIONS - UTILTY

On-line system control and disk management operations are performed by using the commands of the UTILTY system program. UTILTY is a required system program and should be stored in its compiled form in the system library (account [1,2]). While operating with the facilities of the UTILTY system program, the system manager has on-line system control and can perform such operations as:

(a) Enable and disable logins.

(b) Send messages to and force text strings to any or all keyboards on the system.

(c) Terminate execution of a job (kill) or cause a remote line to hang up.

(d) Set and reset system date and time.

The system manager also can perform various disk management operations such as:

(a) Cause private disk packs to be used or to be prohibited from use on specified disk drive units (mount and dismount operations).

(b) Prohibit or allow creation and accessing of files on specified disk drive units (lock and unlock operations).

(c) Rebuild the storage allocation table on a corrupted disk (clean operation).

(d) Change the number of blocks of disk storage an account can retain at LOGOUT time (quota) and change an account password.

(e) Remove all files from a user's account on a specified disk drive unit (zero operation).

The commands of the UTILTY system program are presented in Table 5-2 for reference. The following three sections explain the necessity of the commands according to their functional uses: program control, on-line system control, and disk management.

5.3.1 Invoking and Terminating UTILTY

The system manager or a privileged user executes the UTILITY system program by typing the following command while logged in at any terminal.

```
RUN $UTILTY
```

The program responds by printing one header and one query line as follows:

```
'UTILTY' SYSTEM UTILITY PROGRAM
?
```

after which any valid command can be specified. A list of all valid commands is printed if the HELP command is typed. The query line need not be present in order to type in any subsequent commands. When no commands are pending, the program prints the ? character.

Termination of the UTILTY system program can be properly accomplished by typing either CTRL/Z or the EXIT command. (CTRL/Z is echoed by ↑Z being printed on the keyboard printer.) If CTRL/Z is typed at any time, the operation currently being performed is properly completed; and control is returned to the BASIC-PLUS editor. The completion of the UTILTY run is signalled by READY being printed at the keyboard. If the EXIT command is typed, all currently pending commands are executed; and control is returned to the BASIC-PLUS editor (signalled by READY).

If CTRL/C is typed in order to terminate the program run, any operation in progress is interrupted immediately. Control is returned to the BASIC-PLUS editor. (CTRL/C is echoed at the keyboard printer by ↑C being printed.) Typing CTRL/C to terminate a UTILTY program run is not considered proper termination since the effect of uncompleted internal system operations is unpredictable. The system manager is advised to use CTRL/Z or the EXIT command to terminate UTILTY.

Table 5-2

UTILTY System Program Commands

Category	Command and Format[1]	Use
Operational Control	LOGINS	Enables users to login to the RSTS-11 system.
	NO␣LOGINS	Prevents further login attempts.
	SEND␣{KBn: / ALL}␣xxx	Causes the text string xxx to be printed on the keyboard unit n or all keyboards.
	FORCE␣{KBn: / ALL}␣xxx	Causes the text string xxx to be forced into the keyboard unit n or all keyboards as if it had been typed in.
	KILL␣n	Immediately terminates user job specified by n.
	HANGUP␣KBn:	Disconnects the remote line specified by KBn: from the dataset port.
	DATE␣dd-mon-yy	Sets the RSTS-11 system date to the value of day, month and year (for example, 13-NOV-72).
	TIME␣hh:mm	Sets the RSTS-11 24 hour clock to the value of hours and minutes (for example, 21:52 means 9:52 p.m.).
Disk Management	MOUNT␣dev:id	Logically associates the disk residing on the disk drive unit specified by the device designator dev: with the pack identification label (id) so that data on the disk can be properly accessed by the system. For example, MOUNT DK1:PRIV1 associates the pack PRIV1 with the physical device designator DK1:.
	DISMOUNT␣dev:	Disassociates a disk pack from its physical drive specified by dev:. Must be used prior to removing the cartridge from the disk drive unit.

[1]The notation ␣ indicates that a space character is required.

Table 5-2 (Cont.)

UTILTY System Program Commands

Category	Command and Format[1]	Use
Disk Management (Cont.)	LOCK␣dev:	Places the disk drive unit dev: in a state which prevents files from being OPENed.
	UNLOCK␣dev:	Allows users to OPEN files on disk drive unit dev:.
	CLEAN␣dev:	Rebuild the SAT (Storage Allocation Table) of the pack mounted and locked on disk drive unit dev:. To be used only when message DEVICE NEEDS CLEANING is printed.
	QUOTA␣[n,m]␣q	Sets the quantity of 256-word blocks the user account [n,m] is allowed to retain at logout time to the decimal number q. A value for q of zero means unlimited quota.
	CHANGE␣[n,m] passwd	Alters the password of user account [n,m] to the 6-character alphanumeric passwd.
	ZERO␣dev:[n,m]	Deletes all files from user account [n,m] on the disk drive unit specified by dev:.
Program Control	HELP	Prints a list of valid UTILTY commands at the keyboard printer.
	CTRL/C	Peremptorily terminates execution of the current operation and the UTILTY run.
	CTRL/Z	Allows completion of current operation before termination of the UTILTY run.
	EXIT	Allows completion of pending operations before termination of the UTILTY run.

[1]The notation ␣ indicates that a space character is required.

5.3.2 Principles of Disk Management

Certain commands of the UTILITY system program are used to properly accomplish the management of disks on the RSTS-11 system. Such commands are presented for reference in Table 5-2 under the category of disk management. To more easily convey the proper use of individual disk management commands, the following description of each command is presented with a discussion of its function and value.

5.3.2.1 Preparing a Disk for Use on a Drive - A disk pack or cartridge, to be effectively used on the RSTS-11 system, must be made known to the system. The disk is made known to the system by associating its logical name, the pack label or id, with the physical device unit number on which the disk resides. Such a process is called mounting and is effected by the MOUNT command.

The MOUNT command requires a specific device designator and the exact pack label or identification (id) on the disk which is to be mounted on the system. A new disk is prepared for use on the RSTS-11 system by initialization with the DSKINT program. A disk pack or cartridge, when initialized, is assigned a unique 6-character alphanumeric name called the pack label or id. See Chapter 6 for a description of disk initialization.

When the RSTS-11 system is initialized, at start-up time, only the system disk is mounted. As is described in the discussion of controlling system start-up, the system manager must ensure that other packs in the public structure or private structure are mounted by directives from the START.CTL or CRASH.CTL file used by the INIT system program. The same principle applies to disks added to the system after system start-up. Thus, the system manager mounts a disk on the system by means of the UTILTY system program MOUNT command.

The MOUNT command of UTILTY does not perform two other operations which the MOUNT command in the control files performs and which are necessary before the disk can be safely used. First, the SAT (Sector Allocation Tables) in the SATT.SYS file of the disk device MFD must be rebuilt if necessary. (The message DEVICE NEEDS

CLEANING is printed if the SAT requires rebuilding.) The operation of rebuilding the SAT is called cleaning the disk. The disk must be cleaned if, at the last time the disk was mounted, the SAT being manipulated in core during the file processing, was not placed back in the SATT.SYS file of the disk MFD, thus corrupting the SATT.SYS file. The corruption condition results, for instance, when a user physically removes an operational disk from a drive unit without performing the proper logical dismounting of the disk. When corruption occurs, the amount of storage actually occupied on the disk as a result of file processing is not reflected in the SAT of the SATT.SYS file. Storage occupied is not always reflected in the SATT.SYS file because SAT's (storage allocation tables) are manipulated in core individually and are not written back to the disk to update the SATT.SYS file until the file is CLOSEd.

The system manager, when physically placing a disk in a drive unit, must rebuild the corrupted SAT by using the CLEAN command of the UTILTY system program. Before the clean operation on a disk can proceed, the disk must be in a state which ensures unavailability for normal usage (opening of files). Such a state is termed the locked state. When the disk device is in the locked state, no user except a privileged user can open files on that disk device.

It is therefore necessary, before a clean operation is to be performed, to make the disk device unavailable to users. The system manager makes the device unavailable by use of the LOCK command of the UTILTY system program. After the action of the LOCK command is completed, the system manager then initiates the clean operation by use of the CLEAN command.

Once the clean operation is completed, the system manager can safely make the disk available to users of the RSTS-11 system. He accomplishes this by use of the UNLOCK command. Once a disk device is placed in the unlock state by the UNLOCK command, users are allowed to open files on the disk.

Before a disk pack or cartridge can be properly removed from a drive unit, actions similar to those of preparing the disk for use must be employed. For example, if the system manager desires to replace a private pack with another private pack, he must follow a careful procedure. (Under no circumstances should a public pack be removed from the system during normal system operation.) First, he must ensure that no files are open on the drive unit. The system manager can do this by requesting a disk status report through the SYSTAT system program, which is described in Chapter 4 of the RSTS-11 System User's Guide. His next action would be to lock the device unit by use of the LOCK command. This action ensures that no more files are opened on the disk pack.

When the disk pack to be removed from the drive unit has no OPEN files and has been LOCKed, the system manager next must disassociate that disk pack from the drive unit. He does so by logically dismounting the device with the DISMOUNT command. After the dismounting action is completed, the disk pack can safely be removed from the drive unit. Any pack which is to replace the pack removed must undergo the procedures previously described for proper use of the pack. These procedures are summarized in Table 5-3.

5.3.2.2 Removing Files from an Account - Before an account can be deleted from the RSTS-11 system or deleted from a private pack, the account must contain no files. The ZERO command of UTILTY enables the system manager to remove all files from an account on a device.

5.3.2.3 Changing Quota and/or Password of an Account - Each user account in the RSTS-11 system has associated with it a quota of disk storage that the account can retain at logout time and a password which allows access to the system. The quota and password are specified by the system manager when the account is created.

The system manager can change the quota by use of the QUOTA command of UTILTY. The system manager specifies, in the QUOTA command, the account number and the decimal number of 256-word blocks of disk storage he wishes the account to be able to retain at logout time. If he specifies zero for the quota, he allows the account an unlimited quota of blocks.

The system manager can change the password of an account by using the CHANGE command.

Table 5-3

Procedures for Using Disk Packs

Enter a Pack to the System*	Remove a Pack from the System
1. Use the SYSTAT system program to ensure that the drive unit is free.	1. Invoke UTILTY and use LOCK command on the drive unit containing the pack to be removed.
2. Place the pack in the drive. When the drive is READY, write enable it.	2. Use SYSTAT disk status report to determine the number of OPEN files. If zero, proceed. If non-zero, wait until all files are closed before proceeding.
3. Invoke the UTILTY system program and use MOUNT command to notify system of new pack.	3. With no files open on the device unit, use the DISMOUNT command to notify system that the pack is being removed.
4. Use CLEAN command if necessary.	4. Remove pack from disk drive unit.
5. Use UNLOCK command to free device for use.	

*The disk pack is assumed to have been initialized and formatted using the DSKINT stand-alone program. See Chapter 6 for a description of the DSKINT program.

5.3.3 Operational Control of the System

Certain commands of the UTILTY system program control the operation of the system. Such commands are listed for reference in Table 5-2 under the category of operational control. These commands and examples of their possible usage are described in this section.

When the system manager is on-line in the RSTS-11 system, he can monitor and control system operations. By use of the SYSTAT system program (described in this chapter and in Chapter 4 of the RSTS-11 System User's Guide), he observes the number of free small buffers. If, for example, the number of free small buffers drops below 10, system efficiency declines. He can remedy the possible degradation of system efficiency by preventing more users from logging into the system. The system manager prevents further logins by using the NO LOGINS command. The system manager also uses the NO LOGINS command in preparing to shut down time sharing operations of the PDP-11 computer.

The system manager can communicate with a user at his terminal or with all users by the SEND command. The SEND command causes a specified text string to be placed in the output buffer of a terminal or all terminals and, as a result, be printed on the terminal keyboard printer. If a user assigns a peripheral device for an inordinately long time, for example, the system manager can transmit a message requesting the user to deassign the device. By specifying ALL in place of the device designator of a single keyboard, the system manager can transmit the message to each on-line terminal in the RSTS-11 system.

If it becomes necessary, during the course of system operations, to handle troublesome users, the system manager has two capabilities. He can cause a user's terminal (or all users' terminals) to receive a text string by the FORCE command. He can also terminate a user's job by the KILL command. The FORCE command places a text string in the input buffer of a specified terminal. If the first character of the text is the up-arrow character (↑), it is replaced by a CTRL/C (↑C). The following sequence of two FORCE commands causes a user's terminal to receive two command strings, which, when executed, log the user's job out.

```
?FORCE KB4:  ↑BYE
?FORCE KB4:  YES
```

The KILL command peremptorily terminates a user's job. No files are retained, and none of the normal housekeeping functions are performed. The user is immediately logged off the system.

If the system manager determines that a dataset line is in use but no keyboard activity is taking place (by SYSTAT job status report), he can disconnect the dataset. The HANGUP command causes the remote line specified by KBn: to be disconnected. The hangup capability prevents a user from monopolizing the line without being charged for connect time and frees the line for other remote line users.

The DATE and TIME commands allow the system manager to set the system date and the value of the 24-hour clock, respectively.

5.4 CREATING AND DELETING USER ACCOUNTS - REACT

The system manager or a privileged user creates and deletes system accounts by use of the REACT system program. (The creation and deletion of user accounts can be programmed by a privileged user with the SYS system function FIP codes of 0 and 1, respectively. See Chapter 4 for a description of the SYS system function.) The REACT system program enters user accounts on and deletes user accounts from the system device in the public structure or individual disk devices in the private structure.

The REACT system program should be stored in its compiled form in the system library account [1,2]. REACT is called by using the RUN command as is shown below.

```
RUN $REACT
```

REACT responds by printing the following message which requests that the user specify a function.

```
'REACT' SYSTEM ACCOUNT MANAGER
FUNCTION?
```

The three valid functions are described in Table 5-4, and explained in the following sections.

5.4.1 Creating Individual Accounts - ENTER Function

The ENTER function creates individual user accounts in the MFD on the system device or on a disk device in the private structure. When the system manager runs REACT, he invokes the ENTER function by typing E in response to the request for a function. Upon recognition of the E response, REACT prints a series of questions. A response to each question must be typed by the user before the appearance of the next question. The questions are explained in Table 5-5.

Table 5-4

REACT System Program Functions

Function	Abbreviation	Purpose
ENTER	E	To enter individual accounts on system device in the public structure or a disk device in the private structure.
DELETE	D	To delete individual accounts from the system device in the public structure or from a disk device in the private structure.
STANDARD	S	To create standard user accounts on the system device from the ACCT.SYS file at system generation time.

The following is a sample dialogue for the ENTER function.

```
RUN $REACT
'REACT' SYSTEM ACCOUNT MANAGER
FUNCTION?  E
PROJ,PROG?  100,100
DISK:PASSWORD?  DK1:DEMO
QUOTA?  5000
CLUSTER SIZE?  Ø
PROJ,PROG?  ↑C

READY
```

If the system manager enters an account in the MFD of a disk within the private structure, he permits the owner of the account to create files on that disk. Prior to using REACT to create an account on a private pack, the pack must be mounted and placed in the unlock state by means of the UTILTY system program. Refer to the discussion under the title "Disk Management" in this chapter.

If the system manager enters an account in the MFD of the system disk, he permits the owner of that account access to the RSTS-11 system and allows the owner of that account the use of storage space within the public structure. It is suggested that when a new account is created, the system manager also place an entry for the new account in the ACCT.SYS file. Refer to the discussion of the STANDARD function of REACT.

Table 5-5

Responses to ENTER Function Queries

Question	Response Format	Meaning
PROJ,PROG?	n,m	The user account number to be entered in the MFD, where 1<=n<=254 and Ø<=m<=254.
	CTRL/C	The user terminates the dialogue and REACT by typing the CONTROL key and C combination simultaneously.
DISK:PASSWORD?	passwd	To enter a password for an account on the system device MFD, where password is from 1 to 6 alphanumerics. No value for DISK need be specified since the system disk is assumed.
	device:passwd	To enter a password to an account on a device in the private structure, where device is the device designator and passwd is from 1 to 6 alphanumerics. For example: DK1:PASS DK1 must be logically mounted and in the unlock state by means of the UTILTY program prior to invoking REACT.
QUOTA?	n	The number of 256-word blocks of disk storage the user account is allowed to retain at LOGOUT time where Ø<=n<=65,535 and Ø means no quota is imposed upon the user's account. Therefore, a value of Ø allows the user account unlimited disk storage retention.
CLUSTER SIZE?	n	The account UFD cluster size where n is Ø,1,2,4,8, or 16. If Ø is specified, the pack cluster size is used. If non-zero the value for n must be at least the pack cluster size. Cluster sizes of 1,2, or 4 are recommended for most systems.

5.4.2 Deleting Accounts - DELETE Function

The DELETE function deletes individual user accounts from the MFD of the system device or from the MFD of a disk device in the private structure. As in the case of the ENTER function, a disk device in the private structure must be logically mounted and in the unlock state prior to deletion of an account. In addition, before an account is deleted from an MFD, whether on the system device or on a disk device in the private structure, the UFD of that account to be marked for deletion must contain no files. To clear an account's UFD of files, the ZERO command of the UTILTY system program or the /ZE switch of the PIP system program must be used. See the description of the UTILTY system program in this chapter and of PIP in Chapter 4 of the RSTS-11 System User's Guide.

The DELETE function is invoked by typing D in response to the request for a function. The REACT program prints a series of questions. A response to the question must be typed by the user before the next question appears. The questions are explained in Table 5-6.

Table 5-6

Responses to DELETE Function Queries

Question	Response Format	Meaning
PROJ,PROG?	n,m	The user account number to be deleted from the MFD.
	CTRL/C	The user terminates the dialog and REACT by typing the CONTROL key and C key combination simultaneously.
DISK?	null	By typing the RETURN key, the system device is specified.
	device:	The device designator of a disk in the private structure. For example: DK1: The disk must be logically mounted and in the unlock state by means of the UTILTY system program prior to invoking REACT.

The following is a sample dialog for the DELETE function.

```
RUN $REACT
'REACT' SYSTEM ACCOUNT MANAGER
FUNCTION?  D
PROJ,PROG?  100,100
DISK?  DK1:
PROJ,PROG?  ↑C

READY
```

5.4.3 Automatic Creation of User Accounts - STANDARD Function

The STANDARD function in the REACT system program is present to facilitate automatic creation of a large number of user accounts at system generation time. Explanation of the STANDARD function is presented in detail in two sections of Chapter 2 of this manual: in the overview of system generation section and in the procedural steps for system generation section, under like subsection titles "Establishing System Accounts". A few ancillary remarks are made here.

The ACCT.SYS file is created by PIP as shown in the sample dialog in the procedural steps for system generation section of Chapter 2. The file ACCT.SYS is stored in the system library account [1,2]. ACCT.SYS is an ASCII text file, each line of which is formatted with the following: the items which would be specified by the user in response to the questions of the ENTER function and a name item. Each line of the file represents a single account to be created. The general format is as follows.

```
proj,prog,passwd,quota,cluster,name
```

The items proj, prog, passwd, quota, and cluster are described under the ENTER function. The last item, name, is a user-specified additional designation of the account which is used by the GRIPE system program and which is ignored by REACT. The item, name, must contain no commas, single quotes, or double quotes. The accounts [1,1] and [1,2] can appear in the ACCT.SYS file although they have been created previously during the REFRESH action of the system generation procedure. These account entries in ACCT.SYS are only used by the GRIPE system program.

It is suggested that the system manager devise a program to add the new account information to the ACCT.SYS file each time he creates new accounts by the ENTER function. By updating the ACCT.SYS file, he enables users of new accounts to be identified to the GRIPE system program. The absence of the account information in the ACCT.SYS file does not hinder the operation of the GRIPE program, however.

The system manager can manually enter new account information to the ACCT.SYS file by using the append feature of PIP as follows.

```
RUN $PIP
PIP  -RSTS V4A-12 SYSTEM #213
*$ACCT.NEW<$ACCT.SYS,KB:
1ØØ,1ØØ,DEMO,1ØØ,1,GENERAL USE
↑Z
*$ACCT.SYS/DE
*$ACCT.SYS=$ACCT.NEW/RE
*↑Z
READY
```

5.5 PERFORMING SYSTEM ACCOUNTING OPERATIONS - MONEY

The MONEY system program enables the system manager to extract system accounting information for all accounts in the system or for selected accounts. MONEY is an optional system program and can be run by a non-privileged user to obtain his own account information (excluding password). Refer to Chapter 4 of the RSTS-11 System User's Guide for a description of MONEY for a non-privileged user. MONEY is stored in its compiled form in the system library account [1,2] with a protection of <40>.

MONEY is called by typing the following command while logged into the RSTS-11 system.

```
RUN $MONEY
```

If the caller is a privileged user, a sequence of option queries is printed at the keyboard. Typing an answer to one query causes the next one to be printed. The queries and the explanation for each is given in Table 5-7. The accounting data given as output for each account is presented in Table 5-8.

Table 5-7

MONEY System Accounting Program Options

Option Query	Reply	Explanation
OUTPUT DEVICE?	dev:filename.ext	A file structured or non-file structured device can be specified. For example: LP:
	Typing only the RETURN key	Output is printed at the logged-in keyboard printer.
PRINT PASSWORDS?	NO	Typing anything except NO causes passwords to be printed.
RESET?	YES	Typing anything except YES causes the accumulated accounting data to be preserved. Typing YES causes the following items to be reset to zero. CPU-TIME KCT's CONNECT DEVICE ACCESS
DISK?	DKn:	Type the disk device designator with unit number n to select the accounting data from a private pack.
	Typing only the RETURN key.	The accounting data selected is for all public disks.
SELECTIVE?	YES	Typing anything except YES causes accounting data for all accounts (on the private pack or on the system, whichever the reply to the DISK? query indicates) to be dumped. Typing YES causes an additional query ACCOUNT?

Table 5-7 (Cont.)

MONEY System Accounting Program Options

Option Query	Reply	Explanation
ACCOUNT?	n,m	The account query appears if the reply to the SELECTIVE? query is YES. Accounting data is dumped for the account specified by the project-programmer number [n,m], following which, the query is repeated.
	CTRL/C CTRL/Z	Typing CTRL/C or CTRL/Z terminates the program run.

The MONEY system program can be run during normal time-sharing. No conflict arises if the system attempts to update a user's accounting information while the MONEY is accessing it. When the RESET option is used, MONEY reads and resets to zero the user's accounting information before any system action can update the values being read and reset. Thus, no user accounting information is lost.

Some of the items output can be used to weight billing or evaluate usage. The item KCT, in effect, reflects system usage more accurately than the CPU-TIME item. For example, two users may each exhaust one minute of CPU-TIME in an accounting period. However, one user may tie up 2K words of core while the other may occupy 6K words of core each time he runs. The first user's KCT value is incremented by 2 for each tenth of a second his 2K job occupies core. While the 2K user runs, it is possible for two other 2K user jobs to be running. With the 6K user, each tenth of a second he runs, his KCT value is incremented by 6. The 6K user is tying up more system resources and this is reflected in his higher KCT value. Thus, a user's average job area can be gained by dividing the number of KCT's reflected for the accounting period by the number of tenths of seconds derived from the value of CPU-TIME. Referring to the example of values in Table 5-8, user [100,100] average job size of 3.6K is computed by dividing the number of KCT's, 3000, by the number of tenths of seconds derived from CPU-TIME, 832.

Table 5-8

MONEY System Accounting Program Output

Header Description	Meaning	Example
ACCT	Project-programmer number (account)	100,100
PASSWORD	Account password given at login time	FOO
CPU-TIME	Number of hours:minutes: second.tenths of a second of processor time the account has used since the last reset.	1:23.2 (one minute, 23.2 seconds)
KCT's	Core usage factor (kilo-core-ticks). One KCT is the usage of 1K of core for one tenth of a second.	3000
CONNECT	Number of hours and minutes (hh:mm) of terminal connect time.	2:34 (2 hours, 34 minutes)
DEVICE	Number of hours and minutes (hh:mm) of device usage time, excluding public disks.	20 (20 minutes)
ACCESS	Number of logins and logouts.	8 7 (8 logins, 7 logouts)
DISK	Number of 256-word blocks of disk storage allocated.	100
QUOTA	Number of 256-word blocks the account is allowed to retain at logout time.	500
(no header)	UFD cluster size	2

The value under the header description DISK reflects the actual number of blocks tied up in file allocation on disk. It is not the same value reported by the CATALOG system command. A file may occupy 1 block on a disk as reflected by the CATALOG command, but tie-up 3 additional blocks of disk storage if the file cluster size is 4 blocks per cluster. In essence, the user is depriving the system from claiming those three contiguous blocks in the cluster although the user is not currently occupying the space.

The information given without a header is the UFD cluster size. No other system program returns this value which is determined when the account is created by the REACT system program. It is provided in the MONEY output for information purposes only and has no accounting value.

5.6 MONITORING SYSTEM STATUS - SYSTAT

During normal time-sharing operations, the system manager should monitor the status of the RSTS-11 system. He gains information concerning system status by use of the system program SYSTAT. The SYSTAT system program is stored in the system library account [1,2] with a protection of <168>. The options and output supplied by the SYSTAT system program are described in Chapter 4 of the RSTS-11 System User's Guide. The discussion here gives the system manager guidelines on how and when to use SYSTAT.

Several instances of the use of SYSTAT are described in other sections of this chapter in conjunction with other system manager operations. The previously described instances are listed here.

(a) During preparation for system shut-down, to determine active jobs and disk devices and assignable devices in use.

(b) In conjunction with the UTILTY system program command HANGUP, for determining misuse of a remote line.

(c) In conjunction with the UTILTY command NO LOGINS, when the number of free small buffers is less than 10.

Refer to the discussion relevant to the individual system program or system operation for more information on the above uses of SYSTAT.

Further uses of SYSTAT are listed below and discussed in the ensuing paragraphs.

(d) Monitoring the occurrence of catastrophic errors by the ERROR count.

(e) Uncovering malfunctioning keyboards by the HUNG TTY count.

(f) Guarding against a disk device filling up by watching the FREE block count.

(g) Following the progress of user jobs or detached jobs by the STATE and RUN-TIME items of job status.

Item (d) is important for system managers whose system does not use the crash dump facility. A progressively increasing catastrophic error count indicates some user or users are disturbing system operations or attempting to abuse system facilities. If pre-planned invocation of the ANALYS system program from the CRASH.CTL file is not in effect, no accurate method can be used to determine the guilty job(s). (Refer to the Section 5.2 for a description of the CRASH.CTL file and Section 5.8 for a discussion on analyzing system crashes.)

Each occurrence of a catastrophic error causes the CRASH.SYS file to be written. (That is, if the crash-dump facility was enabled at system start time.) Successive occurrences of catastrophic errors or system crashes cause the CRASH.SYS file to be subsequently overwritten. Thus, running the ANALYS program at random only supplies information relative to the latest system crash or catastrophic error. Therefore, it is recommended that automatic running of the ANALYS system program from the CRASH.CTL file be done. In this manner, automatic monitoring of catastrophic errors is provided. The system manager is kept aware of occurrences of catastrophic errors by the output of the ANALYS system program.

Item (e) refers to the HUNG TTY count reported in the SYSTAT buffer status report. A HUNG TTY count of zero is good. A HUNG TTY count of non-zero indicates the presence of a malfunctioning terminal or terminals. No easy way exists to diagnose the device or devices causing the error count. If the HUNG TTY count appears to increase perceptibly, a field hardware specialist should be consulted.

The FREE block count mentioned in item (f) reflects the apparent number of free blocks in the system and is given in the disk status report of SYSTAT. For practical purposes, however, such as for allocating a file on the device, all of the free blocks reported by SYSTAT may not be usable. A NO ROOM FOR USER ON DEVICE message may be generated although SYSTAT reports enough FREE blocks exist. The cluster size the file uses or the extent of the file in terms of clusters prevents files from fitting on the device desired. For example, a file whose cluster size is 16 and whose length is 10 blocks possibly does not fit on a device which SYSTAT reports to have 50 free blocks of file space remaining. The cluster size of 16 demands that 16 contiguous blocks of free space must exist on the device before the file can be allocated to the device. In some cases, 16 contiguous blocks simply do not exist on a device. (It must be pointed out that RSTS-11 does not allow a file to extend to another physical device.)

A further condition exists for showing NO ROOM FOR USER on a device. The UFD of the user is perhaps full and cannot accommodate the creation of another file. The UFD cluster size was not made large enough when the account was created with REACT.

The occurrence of jobs being stalled in a resource-sharing system is detectable by the means presented in item (g). If the system manager notices that a RUN-TIME value of a job is not increasing (the value is printed out in a job status report), it indicates that the job is stalled, waiting for an I/O device. One user job in the system can ASSIGN a device or keep a device locked by having one file open on it. The system manager can determine the selfish user job by examining the device status report which associates the busy device with the job number of the user controlling that device. The system manager can request that the selfish user free the device or, if that is not viable, can use UTILTY commands to force the guilty job off the system.

The status of detached jobs is of interest also. If a detached job is reported by SYSTAT to be in the HB state (hibernate), it is never eligible for run time. The HB state indicates that the detached job generated an error or has completed execution. The problem of a detached job in the HB state is handled by logging in on a keyboard, by using the attach capability of LOGIN and attaching the job to a terminal. Once the detached job is attached to a terminal, messages can be printed.

5.7 DETERMINING TERMINAL AND REMOTE LINE CHARACTERISTICS - TTYSET AND CONFIG

When the RSTS-11 system is created, the default terminal characteristics of all terminal lines are established as those of the ASR-33 Teletype terminal. During the system build procedures, the system generation program SYSGEN requires that the system manager specify the number of local and remote terminals when giving terminal related parameters. The system manager does not specify the types of local or remote terminals. Since the RSTS-11 system is designed to operate with other type terminals besides the ASR-33 type terminal, two programs are provided to allow the system manager and users of RSTS-11 to take advantage of characteristics of other terminals which are not common to the ASR-33 type of terminal. The two programs are the TTYSET and CONFIG system programs.

TTYSET is the terminal characteristics program of the RSTS-11 system. TTYSET is required on all RSTS-11 systems and must be stored in the system library account [1,2] in its compiled form with a protection of <168>. A description of the capabilities and commands of the TTYSET system program is presented in Chapter 4 of the RSTS-11 System User's Guide.

By design, the effects of the commands of the TTYSET system program are temporary and apply only to the current system initialization period. Because of this ingenious design feature, the type of terminal attached to a keyboard line on the RSTS-11 system is not dependent on the system configuration. The RSTS-11 system operates easily with other types of terminals but is built to expect ASR-33 type terminals on all lines. It is the responsibility of the system manager or the terminal user to specify the characteristics of a terminal different from ASR-33 type terminals in order to take advantage of the characteristics of the different terminal.

For example, the system manager can substitute a terminal with a higher baud rate than an ASR-33 type terminal on a remote line without rebuilding the system. Two methods are available to take advantage of the terminal with the higher baud rate (if the remote line interface has a programmable baud rate). First, the user at a remote site can dial into the RSTS-11 system at the 110 baud rate of the ASR-33 type terminal which RSTS-11 expects. Before beginning time-sharing operations, the remote user can set the characteristics of his terminal by use of the commands of the TTYSET system program. Secondly, the system manager can permanently set the characteristics of that remote line to always respond to the type of terminal with the higher baud rate by use of the CONFIG system program. CONFIG is described in Section 5.7.3.

In another situation, the system manager can replace an ASR-33 type terminal on a local line with a different type terminal. Two methods are available to take advantage of the characteristics of the different terminal. First, the system manager can cause the terminal characteristics to be automatically set each time the system is initialized by use of the TTYSET privileged feature, the KBn: command, from the START.CTL and CRASH.CTL files. See Section 5.7.1 describing the TTYSET privileged feature and Section 5.7.2 describing the method of automatically setting local terminal characteristics each time the RSTS-11 system is reinitialized. Secondly, the user of the local terminal can set the characteristics of the different terminal by use of the TTYSET system program.

As an additional example, the system manager or general user can alter individual, discrete characteristics of a terminal by use of the commands of the TTYSET system program.

5.7.1 TTYSET Privileged Feature - KBn: Command

The system manager sets the characteristics of other terminals in the RSTS-11 system by use of the KBn: command. The following example demonstrates the use of the KBn: command. The system manager, while logged into the system under his privileged account, calls the TTYSET system program as follows.

```
RUN $TTYSET
'TTYSET' TERMINAL CHARACTERISTICS PROGRAM
?
```

The program responds with a header line and a question mark character (?), which indicates that the program is ready to accept commands. If the system manager wishes to set the characteristics of a VTØ5 video terminal at keyboard 3, he types the following commands.

```
?KB3:
?VTØ5
?
```

The question mark character (?) on the third line indicates that the previous KB3: and VTØ5 commands are executed and the program is ready to accept another command for keyboard 3 or another KBn: command.

The system manager can also change the individual characteristics of a local terminal. If he desires to limit the line length of the terminal at keyboard unit 4, he types the following commands.

```
?KB4:
?WIDTH 3Ø
?
```

As a result of the execution of the above commands, whenever 30 characters are printed on a line of the terminal at keyboard unit 4, a carriage return and line-feed operation is performed.

5.7.2 Automatic Setting of Local Terminal Characteristics

The setting of terminal characteristics in the RSTS-11 system applies only to the current system initialization. This condition allows for replacement of an ASR-33 type terminal with any one of the authorized terminals without having to change the system configuration and, thus, rebuild the system. Since it is quite bothersome for the system manager or a user to set characteristics of local terminals each time the system is initialized, an automatic means is provided. (The console terminal must always be an ASR-33 type terminal. Also, characteristics of remote line terminals are permanently set by the CONFIG system program, which is described in Section 5.7.3.)

The automatic means of setting local terminal characteristics is by use of the SET command in the START.CTL and CRASH.CTL files used by the INIT system program at system initialization time. Refer to the sample START.CTL and CRASH.CTL files and the accompanying description of the use of the FORCE directive with the SET command in Section 5.1.

5.7.3 Setting Remote Line Characteristics - CONFIG

A remote line user whose terminal is other than an ASR-33 type terminal must set the characteristics of his terminal each time he logs into the system if he wants to ensure recognition of the characteristics of that terminal by the RSTS-11 system. The characteristics remain set until the telephone is hung up or the line disconnected.

The CONFIG system program is provided in the RSTS-11 system to relieve the remote line user from the necessity of setting the characteristics each time he dials into the system on a certain remote line. The CONFIG system program is optional on the RSTS-11 system and need not be stored in the system library. The program source can be left as distributed in the RSTS-11 kit. When CONFIG is required, it can be read into core by using the RUN command, specifying the device designator of the medium on which it is stored, and the program name, CONFIG. For example, if the RSTS-11 kit is DECtape, the following command is used.

```
RUN DTØ:CONFIG
```

The CONFIG system program responds by typing a query line which requests a remote line number, a comma, and a command, in the following manner.

```
DC11 LINE(Ø-N),COMMAND?
```

On the same line, the system manager types a line number (not a keyboard number); a comma; and a TTYSET command which designates either a distinct terminal characteristic (TAB or NO TAB, for instance) or a set of characteristics for a type of terminal (LA3Ø, for example). The TTYSET commands are listed in Table 4-9 of the RSTS-11 System User's Guide.

The DC11 (or DL11E) line numbers needed in response to a CONFIG query line are not the keyboard numbers. The assignment of keyboard numbers is done on the basis of local line (KL11-type) interfaces and remote line (DC11- or DL11E-type) interfaces. Keyboard number Ø (KBØ:) is always assigned to the console terminal. The lowest keyboard numbers, beginning at 1, are assigned to the local lines. The next highest keyboard numbers are assigned to the DC11-type lines.

The highest keyboard numbers are assigned to the DL11E type lines. The first remote line assigned a keyboard number, the DC11-type line, is line number Ø; the second remote line is line number 1, and so on, until the remote lines are exhausted. These remote line numbers are the numbers to be used to reply to the CONFIG query line.

The query line is repeated after the previous line is executed. When no more remote lines and characteristics are to be given, the program run is terminated by typing the EXIT command in response to a query line.

The following example demonstrates the use of CONFIG.

```
RUN PR:CONFIG
DC11 LINE(Ø-N),COMMAND?  Ø,LA3ØS
DC11 LINE(Ø-N),COMMAND?  EXIT

READY
```

In the above example, CONFIG is run from a DEC supplied paper tape through the high speed paper tape reader (PR:). The system manager specifies that the characteristics of remote line Ø are to be those of the serial DECwriter, LA3ØS. The program run is terminated by typing the EXIT command in response to the next query line. Control is returned to the BASIC-PLUS editor, indicated by the READY message. As a result of the above sample run,the characteristics of remote line Ø are permanently changed to those of the LA3ØS. Whenever the RSTS-11 monitor responds to the ring signal on remote line Ø, it restores the characteristics for that line to those of an LA3ØS serial DECwriter. One caution is in order. The DL11E-type interface does not have programmable baud rates (speeds). Therefore, the user must not use commands to change baud rates on a remote line having a DL11E interface.

5.8 ANALYZING SYSTEM CRASHES - ANALYS

When a system crash occurs in RSTS-11, time-sharing operations are halted. If the required conditions described in Chapter 3 are met, the contents of core are written into the CRASH.SYS file in the system account [0,1] and the system is rebooted into core in auto-restart mode. However, if a catastrophic error occurs, time sharing operations are suspended momentarily. If the No Fail module is present in the system and if the required conditions are again met, certain volatile job information is captured and a request to copy all of core into the same CRASH.SYS file is made and time sharing operations are resumed.

The CRASH.SYS file is dynamic and the information it contains can be lost. The occurrence of a crash or a catastrophic error subsequent to an initial one causes the CRASH.SYS file to be overwritten. Therefore, the system manager is provided with a means of retaining information in the CRASH.SYS file. This means is the ANALYS system program. ANALYS is an optional privileged system program which is stored in its compiled form in the system library account [1,2] with a protection of <168>. The use of the ANALYS system program to document system crashes requires that the CRASH.SYS file be created at system start-up time and that the crash dump feature be enabled at system start-up time.

The ANALYS system program is invoked by the following command:

```
RUN $ANALYS
```

In response, two successive query lines are printed as follows:

```
INPUT?
OUTPUT?  LP:
```

The first query line requests a filename of the file to be analysed, which by default is CORE.SYS in account [Ø,1]. The user need only type the RETURN key, after which the second query line is printed. The second query line requests a device designator for the output medium, which, for example purposes, is the line printer (LP:) in the sample dialogue . Upon completion of the output, program execution is automatically terminated and READY is printed at the keyboard printer.

It is useful for the system manager to cause the ANALYS system program to run automatically. This automatic invocation of ANALYS is accomplished by placing the proper commands in the CRASH.CTL file as FORCE directives. As a consequence of a system crash, the INIT system program, executing the FORCE directives in the CRASH.CTL file, causes a job to be logged into the system, ANALYS to be run, and the job to be logged off the system. The method of accomplishing the automatic invocation of ANALYS is described in Section 5.1.2.

The output of an ANALYS system program run supplies valuable hardware and software information which can be used by a software specialist to determine possible causes of system crashes.

5.9 PROCESSING USER COMMENTS - GRIPE

The RSTS-11 system provides a program to allow users of the system to communicate comments to the system manager. Comments are entered, under the control of the GRIPE system program, to a common file named GRIPE.TXT. The GRIPE system program is optional on the RSTS-11 system and must be stored in its compiled form in the system library account [1,2] with a protection of <168>. The file, GRIPE.TXT, which retains the user comments for inspection by the system manager, is created, expanded, and deleted on an as-needed basis under the system library account [1,2]. As an aid in identifying the user who entered the comment, the GRIPE system program uses an expanded name supplied in the individual user's account information in the ACCT.SYS file, also stored in the system library account [1,2]. However, the expanded name and entry in the ACCT.SYS are not required for GRIPE to run.

5.9.1 User Input Via GRIPE

The user enters comments to a common file for examination by the system manager with the GRIPE system program. GRIPE is invoked by the following command:

```
RUN $GRIPE
```

GRIPE indicates that it is ready to accept user comments by printing a query line as follows:

<u>YES? (END WITH ESCAPE)</u>

The user is then allowed to type the text of his comment which is entered into the common file. The user terminates the text of his comment by typing the ESC or ALT MODE key. (Typing ESCAPE or ALT MODE key is echoed at the keyboard printer by a dollar sign character ($) being printed.) No carriage return-line feed operation is performed. The program indicates its acceptance of the text and its termination by printing the following lines.

```
THANK YOU

READY
```

5.9.2 Privileged Feature of GRIPE

The system manager or a privileged user invokes the GRIPE system program in the same manner as the general user. Once GRIPE prints its query line, the system manager can then examine the contents of GRIPE.TXT or can clear the contents of GRIPE.TXT.

The *LIST command is used in the following manner to examine the contents of the GRIPE.TXT file.

```
RUN $GRIPE
YES?  (END WITH ESCAPE)
*LIST$ OUTPUT? LP:

READY
```

The system manager types *LIST and the ESCAPE or ALT MODE key immediately after the query line. (Typing the ESCAPE or ALT MODE key on a separate line causes *LIST to be entered as text into the GRIPE.TXT file.) If the GRIPE.TXT file is empty, the message NO GRIPES FOUND is printed, followed by the READY message. Otherwise, the GRIPE program requests an output device on which to list the contents of the GRIPE.TXT file by printing the OUTPUT? query. The system manager can type the RETURN key to have the comments listed at the keyboard printer or can type a device designator, such as LP: shown in the example above. The output for each user comment in the GRIPE.TXT file consists of an identification line (including the account entering the comment, the date and time it was entered, and an account name taken from the ACCT.SYS file) and the text of the comment. The program run is automatically terminated upon completion of the output. Control is returned to BASIC-PLUS. This action is signalled by printing of the READY message.

The system manager clears the contents of the GRIPE.TXT file by using the *RESET command after invoking GRIPE. The following example demonstrates the use of *RESET.

```
RUN $GRIPE
YES? (END WITH ESCAPE)
*RESET$

READY
```

The system manager must type *RESET immediately followed by the ESCAPE or ALT MODE key. The clearing of the GRIPE.TXT file and termination of the GRIPE run is signalled by the READY message.

5.10 MANAGING RSTS ERROR LOGGING - ERRCPY, ERRCRS AND ERRDIS

Logging of hardware errors is an automatic function of the RSTS monitor. To gain the advantages of error logging, the system manager must properly employ the ERRCPY, ERRCRS, and ERRDIS system programs.

The ERRCPY program retrieves error-related data logged by the RSTS monitor. Upon occurrence of a hardware error, special routines save the contents of the device registers in memory and transfer the saved data to disk.

The ERRCRS program retrieves error-related data saved following a system crash. When a system crash occurs and certain conditions are in effect, the monitor preserves the contents of certain critical parts of the system. The system file CRASH.SYS holds this information along with other error-related device data. The ERRCRS program transfers the information from the CRASH.SYS file to another disk file which has the same format as the one created by the ERRCPY program.

The ERRDIS system program produces summaries of error-related data and formats it for output to a hard copy device. This program provides the record of errors logged on the RSTS system.

5.10.1 Operation and Use of the Error Copy Program - ERRCPY

The error copy system program ERRCPY reads error-related information stored in the monitor part of memory and writes it to a special disk file. The system manager must ensure that the proper commands are created in the START.CTL and CRASH.CTL files as described in Section 5.1 so that ERRCPY is started and active during time sharing operations. The following discussion outlines the entire process of activating the job which runs ERRCPY.

When the RSTS system starts up, commands in either the START.CTL or CRASH.CTL control file are executed by the INIT system program. If the command FORCE KBØ: RUN $ERRCPY appears in the control file, the command RUN $ERRCPY is placed in the input buffer of the console terminal (KBØ:) as if it had been typed at the terminal. Meanwhile, the accompanying END command in the control file causes termination of the INIT system program and causes the console terminal to be placed at BASIC-PLUS command level (edit mode), as signalled by the READY message being printed. The console terminal remains logged into the system under account [1,2].

When the system executes the command RUN $ERRCPY from the input buffer of KBØ:, the ERRCPY program runs and detaches itself from the console terminal as indicated by the message DETACHING printed at the console terminal. The console terminal is not thereafter logged into the system, but ERRCPY continues running as a detached job under account [1,2].

When ERRCPY is activated, it exists in the SL (sleep) state and neither occupies memory storage nor uses CPU time until awakened by the RSTS Monitor error logging routines. When error logging detects a hardware error, it causes ERRCPY to run and write the error-related information to a special file ERRLOG.FIL. The file is stored under the system library account [1,2] on the system disk. If ERRCPY is not running, the diagnostic area is overwritten and the history of subsequent errors can be lost. Therefore, the system manager must properly start the ERRCPY job.

The system manager gains information concerning the hardware errors detected and placed in the ERRLOG.FIL by running the ERRDIS system program as described in Section 5.1Ø.3. If a system crash occurs, the system manager can retain error data by following the instructions in Section 5.1Ø.2.

5.10.2 Retaining Error Crash Information - ERRCRS

The ERRCRS system program saves error information retrieved at the time of a system crash. When system crash occurs, critical contents of memory are written to the system file CRASH.SYS if the user enabled the CRASH DUMP facility at start up time. The ERRCRS system program transfers certain error information from the file CRASH.SYS to a user designated file. The following sample dialog shows the use of ERRCRS.

```
RUN $ERRCRS
ERRCRS VØ5-Ø6
OUTPUT FILE NAME? FILE.CRS
CRASH DUMP FILE NAME?

READY
```

ERRCRS is executed by typing the RUN $ERRCRS command from a terminal logged into the system under a privileged account. Two queries are printed. The response to the first query designates the name of a file to which error information will be written. The response to the second query is simply the RETURN key, designating the file CRASH.SYS stored under the system account [Ø,1]. The ERRCRS program writes the error information to the file named FILE.CRS (in this sample) and terminates automatically, as signalled by the READY message being printed.

The system manager can later print a report on the error information saved if he uses the ERRDIS system program as described in Section 5.1Ø.3 and designates the filename specified as output of the ERRCRS program run as the input filename for ERRDIS. It is highly recommended that the system manager place the proper commands in the CRASH.CTL file so that ERRCRS runs automatically upon initialization of the system after a system crash.

5.1Ø.3 Operation and Use of the Error Display Program - ERRDIS

The error display program ERRDIS allows the system manager to gain full or partial history or a full or partial summary of the error-related information preserved by the ERRCPY or ERRCRS system programs. ERRDIS prints, in an organized and formatted fashion, the error-related information read from a disk file according to options and switches specified by the system manager. The file is created by either the ERRCPY or ERRCRS system program and exists under the system library account [1,2] with protection code <6Ø>. The disk file can maintain a history of a maximum of 88Ø errors and can record a maximum of 1ØØ of any one type of error. If either of these limits is reached, ERRDIS prints in the output history a message telling how many errors were missed due to no room or to the limit of 1ØØ. The following two sections describe how to run and terminate ERRDIS and how to optimally use ERRDIS features.

5.1Ø.3.1 Running and Terminating ERRDIS - The system manager or privileged user runs the ERRDIS program by typing the following command while logged into the RSTS system.

```
RUN $ERRDIS
```

The program responds by printing a program header line, followed, in turn, on subsequent lines, by three queries as shown below.

```
ERRDIS VØ5-1Ø
INPUT FILE NAME (<CR> FOR DEFAULT)?
OUTPUT TO?
OPTIONS?
```

The user types the RETURN key in response to the query concerning the input file name and ERRDIS prints the second query concerning the output device or file. The default input file name is $ERRLOG.FIL. The user can specify as input the name of the file created by the ERRCRS program. If the user types the RETURN key in response to the second query, the error-related information subsequently requested is printed at his terminal keyboard printer. To indicate a different output device, or file, type the proper specification followed by the RETURN key.

After the user designates the output, ERRDIS prints the OPTIONS query. An option from those given in Table 5-9 can be typed. An option can be modified by any of several switches as described in Table 5-1Ø. After output of the option or options specified is completed, the OPTIONS query is printed again. To terminate the ERRDIS program, type the EXIT command in response to the OPTIONS query.

```
OPTIONS?   EXIT
READY
```

Control is returned to BASIC-PLUS command level, as indicated by the READY message being printed.

Table 5-9

ERRDIS Program Options

Option Type	Option Format	Meaning
General	ALL	Error-related information for all errors is printed in the order in which they were detected and recorded, from the earliest to the most recent.
	EX	Terminate ERRDIS and exit to the monitor.
	HE	Print the help file ERRDIS.HLP.
	MS	Missed errors.
Peripheral Errors	DT	TC11/TU56 DECtape
	RF	RF11/RS11 fixed head disk
	RC	RC11/RS64 fixed head disk
	RK	RK11/RK05 or RK03 DECpack cartridge drive
	CD	Card reader
	MT	TM11/TU10 Magtape
	KB	Hung Teletype errors by job number and keyboard line number.
Processor Errors	T4	Traps through vector location 4
	TØ	Traps through location ØØØØØØ
	JØ	JMP instructions executed to location ØØØØØØ
	RI	Reserved instruction traps
	PF	Occurrences of power failure
	CK	Checksum errors

Table 5-1Ø

ERRDIS Program Option Switches

Switch Format	Meaning
/S	Print only a summary of information of the error type indicated in the option. (Used alone, /S is meaningless.)
/K	Delete (kill) the information from the ERRLOG.FIL file and exit to READY.
/H	Used alone; causes a help file to be printed.
/dd-mmm-yy	Prints information concerning the error type indicated in the option if it was detected on or after the date designated by /dd-mmm-yy. For example, 19-MAY-73.
/hh:mm	Prints information concerning the error type indicated in the option if it was detected at or after the time of day designated by /hh:mm. For example, 8:5Ø or 2Ø:5Ø. If a date switch appears with a time switch, ERRDIS prints errors detected at or after the date and time of day. If a time switch appears without a date switch, ERRDIS uses the current system date.

5.1Ø.3.2 Recommended Usage of ERRDIS - The recommended procedure for using the ERRDIS program is to daily request at least two specific options: ALL/S and ALL. The procedure entails running ERRDIS and answering the OPTIONS query in the following manner.

```
OPTIONS? ALL/S
OPTIONS? ALL
```

The ERRDIS program first creates a summary (/S) of all error-related information. The output starts with 4 lines of accounting data. On the first line, ERRDIS indicates the option requested, followed, on a second line, by the file name from which the information is taken (usually $ERRLOG.FIL). On the third line appears the output specification used and, on the fourth line, the time of day and current date. Following the accounting data is the summary of the total number of errors missed by type.

For the second option requested (ALL), ERRDIS prints the accounting data and the entire history of the errors logged. Information for each occurrence of a logged error is printed in chronological order, beginning with the earliest error and continuing to the most recent occurrence. For each error logged, a header line is printed which describes the type of error and the time of day and date of the occurrence. Following the header line for each error, ERRDIS prints such data as job number, keyboard number (if a hung Teletype error), processor status word (PSW) contents, and the contents (in octal) of the device registers at the time of the error. Consult the PDP-11 Peripherals Handbook for the meaning of the device register abbreviations and the types of errors encountered. (A job number of Ø indicates the null job.) A comment line is appended to some error-related information, such as that of a hung Teletype.

At the conclusion of the error history, ERRDIS prints the number of missed errors (if any) and the total number of errors listed of those logged since the beginning of the error history. It is recommended that the user specify a disk file to contain the output of the options. The printouts of the complete summary and the complete history should be inspected and stored in a central location reserved for them. They provide the basis for planning preventive maintenance and the means to more readily isolate potentially dangerous hardware problems. The printouts should be available to the DIGITAL Field Service or Software Support representative. Periodically, the system manager can delete the contents of the file $ERRLOG.FIL by specifying the following option in response to the ERRDIS program OPTION query.

```
OPTION?  /K
READY
```

The system manager should be alert for certain conditions reported by ERRDIS. Several hung Teletypes are not serious, but a steadily increasing number of hung Teletypes on a certain keyboard line indicates a possibly dangerous condition which should be remedied. Any occurrence of a TØ error is serious and indicates that an interrupting device has presented an incorrect vector location to the bus. An increasing number of disk errors (particularly on the system disk) indicates a need for immediate maintenance.

5.11 SETTING JOB PRIORITY AND TIME QUANTUM FACTORS - PRIOR

The PRIOR system program reports the priority and time quantum factors assigned to an existing job. The system manager can change any of the current values to increase or decrease the chance of gaining run time in relation to other running jobs or to determine how much CPU time the job can have when it is compute bound. The program sets the values by using the -13 FIP call described in Section 4.3.2.

The system runs jobs on the basis of priority. The higher a job's priority, the better are its chances of obtaining run time in relation to other running jobs. Priority is determined by an 8-bit priority byte. By running PRIOR, a privileged user can set the priority for his own job or for another job on the system. Standard priorities are between -128 (lowest priority) and +127 (highest priority). Zero is a legal priority.

When a user first logs in on a system, LOGIN runs with priority Ø. LOGIN automatically sets the user's job to priority -1. This is the default priority with which most or all of the jobs are run. Only in unusual circumstances should priorities other than -1 be assigned.

On occasion, the user may want to run a non-urgent program that requires a great deal of computation. If time is not a factor in obtaining results, the privileged user can decrease the job priority to improve efficiency for the other users on the system. Conversely, infrequently used detached programs often have higher priorities (typically priority Ø) since they can be run quickly once, and ignored thereafter.

The time quantum factors determine the maximum time a job can run compute bound before another job obtains access to the CPU. Each unit of time is equal to 1/6Øth or 1/5Øth of a second, depending on the system's power line frequency. If the system is operating from a 6Ø Hz power line, one unit equals 1/6Øth of a second and six units equal 1/1Øth of a second.

If a specific job is assigned a time quantum of 6 units but does not require that much compute bound time, the system automatically transfers control to the next user before the six units have been used. One tenth of a second is generally considered the best time quantum to insure efficient overall system operation. If a job is guaranteed to become I/O bound (that is, I/O stalled) after a certain amount of computations, PRIOR can be used to specify a time quantum larger than 6. In many cases, a time quantum of 8 units has a significant effect on long computational programs.

5.11.1 Running PRIOR

PRIOR is called as follows.

```
RUN $PRIOR
PRIOR V4B-Ø1 PRIORITY AND QUANTUM CHANGER
```

The first query line printed is:

```
SPECIFY ANOTHER JOB NUMBER?
```

If the current job is to be checked, type the CR key alone. If, however, the job to be checked is not the one under which PRIOR is running, type the job number to be considered. A job number less than 1 or greater than the maximum number of assignable jobs returns the error message ILLEGAL JOB NUMBER ENTERED. Only active, running jobs can be referenced; unassigned job numbers return the above error message.

PRIOR now prints the current priority and time quantum for the job. For example:

```
CURRENT STATISTICS FOR JOB 1:
        PRIORITY                  -1
        ADDITIVE QUANTUM FACTOR    4
        MULTIPLICATIVE FACTOR      1
ANY CHANGES?
```

If any or all of this information is to be changed, type Y in response to the ANY CHANGES query. Typing N or the CR key alone automatically ends the program.

If the user indicates that changes are to be made by typing Y in response to the ANY CHANGES query, PRIOR prints a query line for each parameter in turn. If the value assigned to any parameter is not to be changed, type N or the CR key alone to skip to the next query line. When the typed response to the query line is Y, the message CHANGE IT TO? is printed. Type the new specification as shown below in the sample dialog.

```
ANY CHANGES?  Y
        CHANGE PRIORITY?  Y
        CHANGE IT TO? Ø
        CHANGE ADDITIVE FACTOR?  Y
        CHANGE IT TO? 6
        CHANGE MULTIPLICATIVE FACTOR?  N
CURRENT STATISTICS FOR JOB 1 :
        PRIORITY                  Ø
        ADDITIVE QUANTUM FACTOR   6
        MULTIPLICATIVE FACTOR     1

ANY CHANGES? N

READY
```

Once the last query line has been answered, PRIOR prints the new statistics for verification and prints the message:

```
ANY CHANGES?
```

Typing Y allows the user to make additional changes to this information. Typing N or the CR key alone ends the program.

CHAPTER 6

STAND-ALONE PROGRAMS

This chapter contains the descriptions of two stand-alone programs supplied in the RSTS-11 System. The programs are ROLLIN and DSKINT. ROLLIN program transfers data between DECtape or Magtape and disk devices. The DSKINT program initializes and formats disk storage for use on the RSTS-11 system.

Stand-alone programs are run by using the start-up procedures for RSTS-11 and by answering L as the start-up option. Respond to the query LOAD PROGRAM with the name of the stand-alone program -- ROLLIN or DSKINT. (See Chapter 3 for details on starting RSTS-11.)

6.1 INTRODUCTION TO ROLLIN

ROLLIN is a stand-alone program used to transfer data quickly between a disk and either DECtape or magtape or between RK11 disk cartridges. Disks handled by ROLLIN are the RF11, RC11, RPØ2, and RK11. ROLLIN assumes no file structure on disk or DECtape; transfers are performed in image mode. Magtapes are treated as file-structured devices in that each ROLLIN file is preceded by a DOS-compatible *file label* (see paragraph 6.1.4).

When transferring data onto DECtape or magtape, ROLLIN automatically writes an initial record containing a tape sequence number called a *reel label*. For DECtape transfers, the reel label also contains the number of blocks of data transferred. The reel label guards against mounting tapes out of sequence when returning data to a disk device.

Preceding all data records on a DECtape or the first file on a magtape, ROLLIN copies a core image of itself. This image permits ROLLIN to be bootstrapped off DECtape or magtape to load the remainder of the tape.

6.1.1 Command Format

ROLLIN prints a # character on the teleprinter (or console teleprinter for a multiple terminal system) whenever it is ready to accept a command. ROLLIN commands are of the form:

```
#dev1:<dev2:/options
```

or, more simply:

```
#/options
```

where:

dev1:	is the output device specification
dev2:	is the input device specification
/options	indicates one or more legal switch options

The # character is underlined in examples to indicate that it is printed by ROLLIN rather than typed by the operator.

Legal device specifications are as follows:

Device Specification	Device
DF:	RF11 disk (256K unit)
DC:	RC11 disk (64K unit)
DKn: (n = Ø to 7)	RK11 disk, units Ø to 7 (1.2 million words/cartridge)
DPn:	RPØ2 disk, units Ø-7 (10 million words per disk pack)
DTn: (n = Ø to 7)	DECtape, units Ø to 7 (each reel contains 578 blocks, each 256 words long)
MTn:name (n = Ø to 7)	Magtape, units Ø to 7 (A filename must be specified. It is written in the *file label* and must not include a filename extension)

The digit Ø should not be omitted from the specification when a unit is indicated. For example, DK: is illegal.

An example of a ROLLIN command is shown below:

```
#MTØ:DAT8<DC:
```

The above command dumps the entire contents of an RC11 disk onto a magtape file called DAT8 on the magtape reel mounted on magtape unit Ø.

TABLE 6-1

ROLLIN Option Switch Descriptions

Option Switch	Abbreviation	Meaning
/BOOT:dev	/BO:dev	This general-purpose bootstrap switch, which loads the first 256 words of core from the device and then branches to location Ø, is used to bootstrap DOS or RSTS; for example: #/BO:DK loads and starts an RK11 DOS system.
/DATE:date	/DA:date	Used in writing the FILE LABEL on magtape. /DA causes the specified date to be entered in the FILE LABEL. The format for the date is day-month-year; for example: #/DATE:1-FEB-72
/FIND	/FI	Used with magtape only. When reading from magtape, /FI rewinds the tape and searches for the specified file. When dumping to magtape, /FI causes the tape to skip past all the files previously written on the tape.
/HELP	/HE	Types a brief explanation of ROLLIN options on the console.
/NOLABEL	/NO	On dumping disk to DECtape, this option inhibits the *reel label* record from being written.
/NUMBER:n	/NU:n	n is the number of 1K disk tracks to dump or restore. If /NUMBER is not specified, the entire disk is used.
/PLATTERS:n	/PL:n	For use with RF11 or RC11 disks only, n is the number of disk platters to load or dump./PLATTERS:1 is always assumed if not specified.
/RSTS	/RS	This special-purpose bootstrap switch is used to bootstrap RSTS from the RF11 disk.

TABLE 6-1 (Cont.)

ROLLIN Option Switch Descriptions

Option Switch	Abbreviation	Meaning
/RWIND	/RW	When reading or writing magtape, this switch causes the tape to be rewound before use.
/SKIP:n	/SK:n	When reading or writing magtape, this switch causes the tape to skip past n end-of-file marks before starting a read or write.
/V4A	/V4	Obsolete option.

6.1.2 Editing Commands to ROLLIN

Editing procedures for ROLLIN command strings are identical to those for DOS and RSTS command strings.

While typing a command line, the operator can use the RUBOUT key repeatedly to delete preceding characters. The entire command line can be deleted by typing CTRL/U.

When ROLLIN is printing on the the teleprinter, typing CTRL/O inhibits further printing.

6.1.3 Disk/DECtape Operations

DECtape is a block-structured device. DECtape reels contain 578 blocks, each 256 words in length. When ROLLIN writes a DECtape, it dumps its own image onto blocks Ø to 12, writes a label onto block 15, then dumps data from the disk onto blocks 16 through 527. This means that 128K of data can be kept on each tape. Hence, an RF11 disk requires two DECtapes, while an RKØ3 disk requires ten DECtapes for a complete dump.

When a disk is restored from DECtape, ROLLIN checks the label block on each reel to ensure that the tapes are used in the proper sequence, and that the number of blocks restored does not exceed the number of blocks dumped. If this is not the case, an error message is printed.

6.1.3.1 Disk to DECtape Dump

To dump disk to DECtape, simply type DT: as the output device and the disk name as the input device. For example:

```
#DT:<DKØ:
```

would dump RKØ3 disk cartridge Ø onto ten DECtapes (unit Ø would be used first, then units 1, 2, 3, 4, 5, 6, and 7, the ninth tape would then go back to unit Ø and the tenth to unit 1). This is the normal order for using tapes; the user could override this by typing the actual units to use. For example:

```
#DT2:,DTØ:<DF:
```

would dump the RF11 disk onto two tapes, using unit 2 first, and then unit Ø.

The use of the /NOLABEL option requires a brief explanation. This would only be used when writing over the first part of a DECtape previously written by ROLLIN. For example, suppose we have just patched the RSTS Monitor. This Monitor normally resides on the first 28K of the RF11 disk. Now suppose we want to dump this patched Monitor onto the first 28K of an earlier disk dump. This can be done by mounting the first dump tape on unit Ø, loading ROLLIN, and typing the following command:

```
#DTØ:<DF:/NUMBER:28/NOLABEL
```

This will write out the first 28K of the disk without rewriting the DECtape label. Now, when we restore from this tape, ROLLIN will read the entire tape, and not just the first 28K.

6.1.3.2 Restoring Disk from DECtape

This is simply the reverse of the dump operation and takes a similar command. For example:

```
#DF:<DT:
```

would restore the RF11 from two reels of DECtape on units Ø and 1.

As before, reel numbers can be given explicitly, if desired; for example:

```
#DF:<DT1:,DT2:,DT3:,DT4:/PLATTERS:2
```

would load a 2-platter RF11 disk (512K) from four DECtapes, mounted on units 1, 2, 3, and 4.

6.1.4 Disk/Magtape Operations

Magtapes are written by ROLLIN with 8ØØ BPI density and odd parity. On 9-channel drives each 16-bit word takes two frames of tape. On 7-channel drives core-dump mode is used, so each 16-bit word takes four frames of tape.

Each dump operation writes one file on the tape. This file normally consists of a 7-word file label record containing the filename (always with the .ROL extension), UIC (always [1,1]), protection code (always 155), and date. This is followed by a 256-word reel label record that contains the reel sequence number for this file. Following this are the actual data records. Each data record, except possibly for the final one, contains 4K words (8K bytes) of data. ROLLIN closes the file by writing three end-of-file (EOF) records, one to end the file and two more to indicate the end-of-data (EOD) on the magtape. ROLLIN then backspaces over two of the EOF records to leave the tape correctly positioned for performing another dump. There is one exception to this file format. If, before writing the file label, the tape is positioned at the load point (beginning-of-tape), two records are written immediately following the file label. These two records contain a bootstrap and an image of ROLLIN, and they are used in conjunction with the routine to load ROLLIN off magtape.

The file label is DOS compatible, permitting mixing of ROLLIN files and DOS files on a single tape and cataloging of the tape via the /DI option of DOS PIP. However, DOS records are required to be 256 words in length, while ROLLIN makes more efficient use of the tape by writing 4K words in each record. This means that ROLLIN files cannot be read in file structured mode by DOS (READ/WRITE) TRAN can, of course, be used with ROLLIN files (see the DOS Monitor, Programmer's Manual).

Figure 6-1 illustrates the format of a tape with two files on it, the first of which is a dump of an RF11 disk, and the second a dump of an RKØ3 disk cartridge.

6.1.4.1 Disk to Magtape Dumps

To illustrate dumping from disk to magtape, suppose a system had one RF11 disk and two RKØ3 drives. These could be dumped as three files by the following commands:

```
#MTØ:DFDMP<DF:/RWIND/DATE:1-FEB-72
#MTØ:DKØDMP<DKØ:/DATE:1-FEB-72
#MTØ:DK1DMP<DK1:/DATE:1-FEB-72
```

The filename must be specified with magtape, the date is optional but should be specified. The magtape now has three files on it: DFDMP, DKØDMP, and DK1DMP. Note that the file extension is *never* specified by the user; the extension will always be a .ROL.

This assumes that a tape has been mounted, WRITE ENABLEd, on unit Ø. To use another unit simply designate MT1:, MT2:, etc. in place of MTØ:.

A large 2400-foot reel can easily hold this much data; however, a small 600-foot reel would not have enough room for a complete dump of the second RKØ3. When ROLLIN detects the end-of-tape, it backspaces over the last data record, writes three end-of-file records, and prints the following message:

```
TAPE FULL, TYPE M TO MOUNT ANOTHER REEL AND CONTINUE,
ANYTHING ELSE TO ABORT REQUEST:
```

If another tape is available, type M followed by the RETURN key. ROLLIN now prints the following message:

```
TYPE RETURN TO CONTINUE WHEN READY.
```

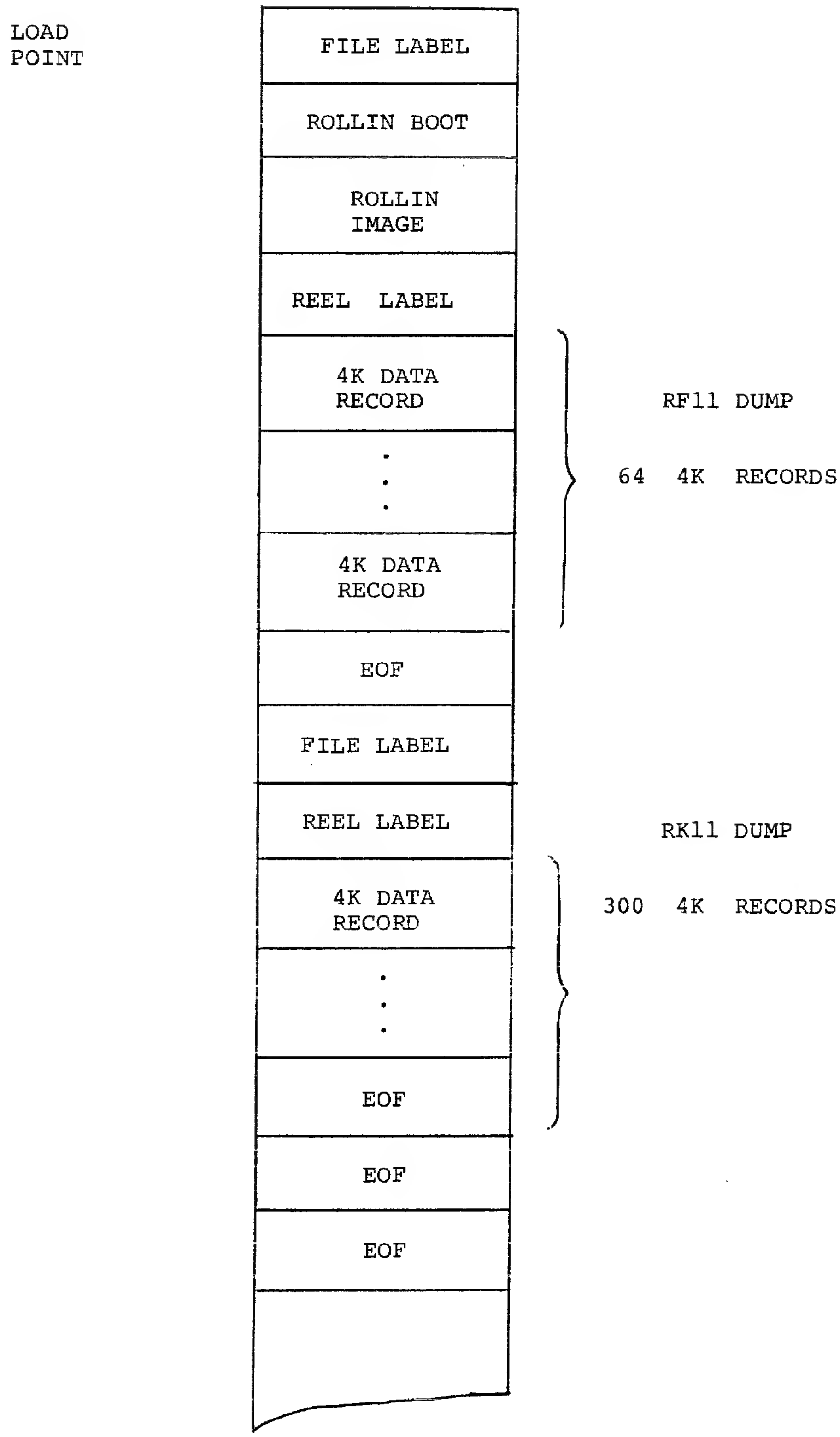


Figure 6-1 ROLLIN Magtape Format

Now mount the new tape (it must be on the same unit number). Type the RETURN key and ROLLIN will finish the dump.

After a tape has been dumped it is recommended that the WRITE ENABLE ring be removed.

6.1.4.2 Restoring Disk from Magtape

Returning to our illustrative case, the three files could be restored from magtape by the following commands:

```
#DF:<MTØ:DFDMP/FIND
#DKØ:<MTØ:DKØDMP
#DK1:<MTØ:DK1DMP
```

Here, the /FIND switch rewinds the tape and searches for the file DFDMP which would in this example be the first file on the tape. While /FIND may always be used, if you are certain that the tape is correctly positioned, omitting /FIND will greatly speed the restore process since no search for the file will made. ROLLIN will, in any case, verify that the filename is the same as the specified name, and that the extension is .ROL, and give an error if it is not.

This again assumes that the tape dumped by the commands given in Section 6.1.4.1 has been mounted on unit Ø. If an end-of-file is encountered during the read, as would be the case with a 600-foot reel, then ROLLIN will ask for another reel by printing:

```
END-OF-FILE DURING READ, TYPE
M TO MOUNT ANOTHER REEL, OR K TO KILL REQUEST:
```

If the second reel is available, respond M followed by the RETURN key. ROLLIN now prints the following message:

```
TYPE RETURN TO CONTINUE WHEN READY.
```

Now dismount the current tape and mount the continuation tape (it must be on the same unit number, as before). Type return and ROLLIN will finish the read.

6.1.5 Copying RK11 Cartridges

One more option available with ROLLIN is to do a dump from one RK11 cartridge to another. In this case, a direct copy is done, with no header or labeling information being written. For example:

```
#DKØ:<DK1:
```

would copy the entire contents of the cartridge mounted in drive 1 onto the cartridge mounted in drive Ø.

6.1.6 Error Messages

Error messages promulgated by the ROLLIN program are succinct and require little interpretation by the operator.

SYNTAX ERROR, COMMAND IGNORED.

DISK ERROR,---REQUEST KILLED.

THE REEL LABEL INDICATES THAT THE REST OF THE TAPE WAS NOT DUMPED, TYPE K TO KILL REQUEST AT THIS POINT, ANYTHING ELSE TO PROCEED IN THE FACE OF DANGER:

LABEL INDICATES THAT THE TAPE IS OUT OF SEQUENCE. TYPE P TO PROCEED, M TO MOUNT ANOTHER REEL, OR K TO KILL REQUEST.

PREMATURE END-OF-FILE, REQUEST KILLED.

TAPE FULL, TYPE M TO MOUNT ANOTHER REEL AND CONTINUE.
ANYTHING ELSE TO ABORT REQUEST:

MAGTAPE SELECT ERROR.

MAGTAPE WRITE PROTECT ERROR.

FATAL MAGTAPE ERROR.

SPECIFIED DEVICE DOES NOT EXIST.

REACHED END-OF-DATA ON SKIP, OPERATION KILLED.

HUNG DEVICE DTn
TYPE K TO ABORT, ANYTHING ELSE TO TRY AGAIN:

MOUNT TAPE ON DECTAPE n
TYPE RETURN TO CONTINUE WHEN READY.

TOO FEW DECTAPE UNITS WERE SPECIFIED. REQUEST KILLED.

6.2 INTRODUCTION TO DSKINT

DSKINT is a stand-alone program used to format RK11 or RPØ2 disks and to build RSTS-11 structures on RK11, RPØ2, or RF11 disks. The structures built by the DSKINT program are the MFD, the [0,1] UFD, the Monitor files SATT.SYS and BADB.SYS and a dummy bootstrap routine. The format and building operations permit the initialization of RSTS-11 private disk, public non-system disks and enables bad block checking of all RSTS-11 disks.

6.2.1 Running DSKINT

To run DSKINT, follow the starting procedures of Section 3.2. Request the LOAD option and proceed as described in Section 3.3.3. When DSKINT is loaded, the RSTS-11 Monitor is overlayed and destroyed. Therefore, the program re-initializes itself after completion of each run.

The user is informed that DSKINT is ready to run by the following header and query lines printed at the console:

```
RSTS DISK INITIALIZER V4A-ØØ

DISK(RF,RK,RP) =
```

The user supplies the proper choice of disk types from one of those shown. RSTS-11 does not check if the device is legal for the system. It checks for the proper two characters only.

If any typing errors are made, typing the RUBOUT key echoes back the last character typed and removes it from the line buffer. If RUBOUT is typed once for each character entered, the line buffer is emptied and further occurrences of RUBOUT are ignored. Typing CTRL/U allows retyping of the current line. Typing CTRL/C restarts the DSKINT program. LINE FEED and RETURN characters are interchangeable.

The dialogue continues at the console until the user is requested to ready the disk unit and type the RETURN key. If no errors are detected, the termination dialogue is signalled by the following lines being printed on the console:

```
INITIALIZATION COMPLETE

RSTS DISK INITIALIZER VØ4A-ØØ

DISK(RF,RK,RP) =
```

6.2.2 Terminating DSKINT

The DSKINT program is terminated by rebooting the RSTS-11 Monitor and following the start procedures described in Chapter 3.

6.2.3 Errors in Dialogue Queries

The DSKINT program detects invalid user entries during the dialogue, prints "BAD SPECIFICATION", retypes the question to which the invalid response was typed. This can happen in any of the following cases:

1) DISK NOT RF, RK, OR RP
2) DRIVE NUMBER GREATER THAN 7
3) PACK CLUSTER NOT 1, 2, 4, 8, OR 16
4) MFD CLUSTER NOT GREATER THAN OR EQUAL TO PACK CLUSTER OR NOT 1, 2, 4, 8, OR 16
5) MFD OR PACK ID NOT ALPHANUMERIC
6) PUB OR PRI NOT SPECIFIED

6.2.4 Errors in Formatting Operations

If the specified disk unit is WRITE PROTECTED or NOT READY the following line is typed:

```
DISK NOT READY - RESTART
```

The DSKINT dialogue is restarted.

If, for any other reason, the disk cannot be formatted, a disk diagnostic is printed (with NA for "not applicable" replacing the sector number) followed by the message:

```
FORMATTING FAILURE
```

The DSKINT dialogue is restarted. (See Section 6.2.6 for disk diagnostics.)

If the disk has not been formatted and the user did not answer Y to the formatting question, a diagnostic is generated for each sector on the disk. If this occurrence is evident, the user should type CTRL/C to restart the dialogue rather than wait for a diagnostic to be printed for each sector on the disk.

6.2.5 Errors in Initialization Operations

If an error occurs during creation of the directories, the Monitor files, or the bootstrap routine; or if a hardware disk error cannot be cleared; a disk diagnostic is typed, followed by the message shown below:

```
FAILURE - UNRECOVERABLE ERROR DURING FINAL INIT.
```

The dialogue is restarted. (See Section 6.2.6 for disk diagnostics.)

If the number of sectors which are found to be bad exceeds 256 times the pack cluster size, the following message is typed:

```
EXCESSIVE BAD SECTORS
```

The user should not wait for all of the disk diagnostics to print at the terminal if it is clear that a diagnostic is being printed for each sector. For example, to generate an EXCESSIVE BAD SECTORS message for a pack cluster size of 1, 401 (octal) sector diagnostics are printed at the console. The printing time is over twenty minutes.

If one of the sectors allocated to the directories, the Monitor files, or the bootstrap routine is found to be bad, the following message is typed:

```
INIT FAILED - AT LEAST 1 CRITICAL SECTOR IS BAD
```

The dialogue is restarted.

6.2.6 Errors on the Disk

Errors on the disk are reported by sector number (an octal value) and the related contents (octal) of selected hardware registers, dependent upon the type of disk device. Upon the first occurrence of a sector being determined to be unusable, a title line is printed; the title line contains mnemonics describing sector and which hardware values are placed in that column. Although some of the mnemonics appear different in spelling, the PDP-11 values have not changed. For example, RKMA is equivalent to RKBA, where M (memory) has replaced B (bus) in the bus address register.

6.2.7 Execution Times by Disk Type

The following values are approximate times for formatting and initializing the various types of disks. The values should be used as a guideline only.

Operations	Device Type		
	RF11	RK11	RP11
Formatting	Not allowed	30 seconds	1 min, 30 secs.
Initialization	1 minute	7 minutes	35 minutes

APPENDIX A

SYSGEN INTERROGATION SUMMARY

QUESTIONS	RANGE OF ANSWER	DEFAULT ASSUMPTION FOR NULL ANSWER
Long or short form of questioning?	S for short form. Anything for long form.	Long form
System name?	Up to 15 alphanumeric characters.	(none)
AC power frequency?	6Ø, 5Ø	6Ø
Line clock or programmable clock?	L, P	Line Clock
Power Fail code?	Y, N	Y
Highest job number?	1 through 17	(none)
Number of KL11 interfaces?	Ø through 16	Ø
Number of DC11 interfaces?	Ø through 16	Ø
Number of DL11E interfaces?	Ø through 16	Ø
Terminal "fill" capability?	Y, N	N
Serial LA3Ø?	Y, N	N
Lower Case	Y, N	N
Escape	Y, N	N
XON/XOFF remote reader control?	Y, N	N
Automatic DC11 answering?	Y, N	N
CRT terminals?	Y, N	N
No parity(1), even parity(2), or odd parity(3) for terminal output?	1, 2, 3	No parity generation (1)
Number of RF11 disk platters?	Ø through 8	1
Number of RK11 disk drives	Ø through 8	Ø

QUESTIONS	RANGE OF ANSWER	DEFAULT ASSUMPTION FOR NULL ANSWER
Swap only RF	Y, N	N
Number of RC11 disk platters?	Ø through 4	Ø
High speed reader?	Y, N	N
High speed punch?	Y, N	N
Number of dual DEC-tape drives?	Ø through 4	1
Number of big buffers	1 through 8	2 times the number of dual DECtape drives.
Card reader?	Y, N	N
Card reader decode table: Ø29, Ø26, or 14Ø1?	Ø29, Ø26, 14Ø1	Ø29 decode table
Line printer?	Y, N	N
LS11 type printer?	Y, N	N
Line printer vertical format software?	Y, N	N
Number of print-columns on line printer?	8Ø, 132	8Ø
96 character set on line printer?	Y is 96; N is 64	N (upper case only)
Number of MagTape drives?	Ø through 8	Ø
Number of Small Buffers?	1Ø through 999	64
Math package?	2, 2E, 2X, 2EX, 4, 4E, 4X, 4EX	(none)
24-hour time	Y, N	N
Catastrophic error recovery?	Y, N	N
Record I/O	Y, N	N
Update?	Y, N	N
PRINT USING?	Y, N	N
Matrix verbs?	Y, N	N
Line printer assembly listings?	Y, N	N

QUESTIONS	RANGE OF ANSWER	DEFAULT ASSUMPTION FOR NULL ANSWER.
Media	DT, MT, DK	(none)
DSKINT in CIL	Y, N	N
ROLLIN in CIL	Y, N	N

APPENDIX B

RSTS-11 TERMINAL SOFTWARE OPTIONS

B.1 ESCAPE, ALTMODE, PREFIX

Some terminals have outmoded ASCII character keys:

175_8 = ALTMODE

176_8 = PREFIX

More recently designed terminals incorporate the 1968 ASCII character set, in which the following control characters appear:

33_8 = ESCAPE

175_8 = }

176_8 = ~

RSTS-11 interprets 33_8 as a line terminator. On input, RSTS-11 software will also automatically translate 175_8 and 176_8 into 33_8 and consequently treat them also as line terminators. Depending however, on the option chosen at system-generation time, the system program TTYSET can be used to alter internal parameters for a given terminal so that 175_8 and 176_8 are not translated but remain themselves.

B.2 LOWER AND UPPER CASE CHARACTERS

Some terminals can send and can print both lower-case and upper-case characters. Thus, when a lower-case character is sent, the terminal is capable of printing the echo in lower case so that the user has an accurate visual representation of his lower-case transmission.

Other terminals (e.g., VTØ5) can send either lower-case or upper-case characters but can print only upper-case characters. Whenever they receive a lower-case character, they print the corresponding upper-case character. Consequently if a user sends a lower-case character, the echo is printed in upper case and the user has no clear visual indication that his transmission was indeed in lower case.

Other terminals, (e.g. ASR33, KSR33), can neither send nor receive lower-case characters. If they receive a lower-case character, they will print the corresponding upper-case character.

To protect users of the second type of terminal from inadvertently sending lower-case characters into the system or into a data file, RSTS-11 software can be configured with the capability of automatically translating all input lower-case characters to their corresponding upper-case counterparts before processing them. When this capability has been configured into the system, TTYSET has the power of selectively enabling or disabling it for any specified terminal.

B.3 GENERALIZED FILL

At system-generation time, the RSTS-11 system can be configured to have the generalized terminal fill feature. If this feature is not included in the configuration, no fill-characters are sent by the RSTS-11 system. If this feature is included in the configuration, RSTS-11 is capable of automatically sending a sequence of nulls ($Ø_8$) as fill-characters after certain control characters. The transmission of these meaningless nulls allows the terminal sufficient time to complete the physical action initiated by the control character and to become synchronized or ready to receive and print the next meaningful character. The control characters after which the null characters are sent and the base number of null characters which are sent after each is shown in the table below.

Control character to be followed by null(s)	Base number of nulls which will follow
<CR> (15_8)	1
<TAB> (11_8)	1
<VT> (13_8)	4
<FF> (14_8)	9

The actual number of null characters that are sent after a control character are the base number indicated in the table multiplied by a fill factor set by TTYSET. For example, if a user runs TTYSET and sets a fill factor of Ø for his terminal, then no fill characters are issued after any control character. If, however, he sets the fill factor for his terminal to 2, then 2 nulls will be output after <CR> or <TAB>, 8 nulls after <VT>, and 18 nulls after <FF>.

B.4 XON/XOFF REMOTE READER CONTROL

If a low-speed paper-tape reader on a remote terminal (connected to the system through datasets and communications lines) is to be used under RSTS-11, three requirements must be fulfilled:

(1) the terminal must be equipped with the requisite hardware for XON/XOFF remote reader control,

(2) the RSTS-11 software must be configured at system-generation time to include the XON/XOFF software for remote reader control, and

(3) TTYSET must be used to enable the XON/XOFF code for the given terminal.

For local terminals with or without low-speed readers or for remote terminals without low-speed readers, none of this is required.

B.5 OUTPUT PARITY BIT

RSTS-11 software can be configured at system-generation time to add always an odd or an even parity bit to its output to terminal-devices, or it can be configured to omit always the parity bit.

DEC-supplied terminals ignore parity bits and operate in the same fashion whether or not RSTS-11 sends them a parity bit. This may not be true of non-DEC terminals.

B.6 AUTOMATIC ANSWERING FOR REMOTE TERMINALS

RSTS-11 can be configured at system-generation time to include an automatic answering facility for remote terminals connected to the system through dataphone ports. When this feature is included in the software configuration, the following sequence of actions occur:

(1) Remote user dials and rings at the RSTS-11 dataset port.

(2) RSTS-11 sends its carrier to the remote user.

(3) The remote dataset sends its carrier back to RSTS.

(4) Upon reception of the remote terminal carrier, RSTS-11 simulates an I<LF> as being received from the remote user. This will cause the system program LOGIN to run.

(5) LOGIN will scan the user's input buffer and find the I. Upon detecting the I, LOGIN branches to an automatic answering routine. This routine as it is found on the supplied LOGIN.BAS causes the same action to be taken as happens when HELLO is found in the input buffer. The system manager, however, may wish to alter the automatic answering routine in LOGIN to perform whatever task or send whatever message he wishes.

When the automatic answering software is excluded from the configuration, steps (4) and (5) never occur.

APPENDIX C

SYSLOD ERROR MESSAGES

The system loader program SYSLOD transfers a linked core image library from one device to a disk as a contiguous core image library. The user runs SYSLOD from magtape or DECtape during the system generation procedure. The user must consult this section for error messages and possible recovery procedure if errors are generated during SYSLOD execution.

Error messages issued by the SYSLOD program can be of two types: recoverable errors and non-recoverable errors.

C.1 RECOVERABLE ERRORS

The following errors are diagnosed and printed by SYSLOD. Once the error message is printed, SYSLOD restarts by identifying itself again, and printing the # (input request) character at the keyboard. The user should retry the most recent command, making the indicated corrections. Error messages for recoverable errors are preceded by one of the following:

CIL dev LICIL dev	(dev represents the device mnemonic)

depending upon whether an error has been detected in the CIL (output) or LICIL (input) side of the most recent command string.

SYNTAX ERROR

This message is printed if the command input line contains a syntax error.

TOO MANY SWITCHES

SYSLOD does not accept switches on the input side of the command string. If too many switches are specified on the output side of the command string for SYSLOD to handle, it issues this message.

UNKNOWN SWITCH

If SYSLOD does not recognize the switch as a valid switch, it prints this message.

SWITCH ERROR

If a switch is used incorrectly, SYSLOD prints this message. Incorrect use of a switch implies specification of an argument when no argument is valid or the lack of an argument when one is required.

SWITCH CONTEXT ERROR

This message is issued when switches are specified incorrectly for their definitions.

ERROR IN SWITCH ARGUMENT

This message is issued when decimal argument for any switch is too large to be contained in 16 bits.

NONEXISTENT DISK OR DISK NOT READY

Either (1) the disk specified in a command string does not exist in the configuration, or (2) the disk exists, but is not ready.

UNKNOWN DISK NAME SPECIFIED

This message is issued when a disk name other than those listed below is specified in a command string.

DF (DFØ through DF7)
DK (DKØ through DK7)
DP (DPØ through DP7)

ERROR WHILE FORMATTING RK DISK

This message is issued whenever an error is detected while formatting an RK disk unit.

A xxx
READY

This message is issued when a problem exists with a peripheral device not in the "ready" state. Check the device and the operating instructions to determine the cause of the error and correct the error. When the problem has been rectified, type YES and the CARRIAGE RETURN key to continue.

C.2 NON-RECOVERABLE ERRORS

The SYSLOD program prints an error message at the keyboard device when a non-recoverable error is encountered during processing. These messages are listed below, along with the action to be taken (if any).

INPUT IS NOT A LICIL

The first line of the input file is not in correct format for a LICIL. The most probable cause of this error is an attempt to transfer a load module (filnam.LDA) instead of the LICIL (filnam.LCL).

END OF FILE BEFORE CIL LINE READ

This error message is issued when SYSLOD reaches end-of-file before detecting the CIL line. If the CIL is being loaded from DECtape, magtape, or disk, it is likely that part of the file has been destroyed and must be rebuilt.

BOOTSTRAP NOT IN BLOCK Ø

This message occurs in Replace mode only (neither the NS nor ZE switch has been specified). In Replace mode, SYSLOD searches bootstrap parameters to find the CIL to be replaced. If the first block number of the CIL hooked to the bootstrap (location 176 of BOOT) is Ø, then block Ø is not a hooked bootstrap, and this message is issued.

BOOTSTRAP NOT HOOKED TO CIL; CANNOT REPLACE

This message occurs in Replace mode only; (neither the NS nor ZE switch has been specified). If the first block indicator of the bootstrap (location 176 of BOOT) is non-zero, it must be pointing to a CIL. If the first formatted binary line of the "hooked" file is not COMD section #3, then the file is not a CIL.

BLOCK SIZE DISCREPANCY BETWEEN CILUS AND SYSLOD

This error message is issued when the NS switch is used with SYSLOD. Exactly the same parameters must be specified for SYSLOD as were specified for CILUS.

LICIL TOO BIG, NOT ENOUGH RESERVED BLOCKS

This message occurs for either of two conditions:

a) In Replace mode, the new LICIL is larger than the old.
b) In any other mode, the number of reserved blocks (BL:nnnn) is not large enough for the CIL.

1ST LINE NOT COMD SECTION #4 OR 1

After reading the CIL line, SYSLOD begins to load the LICIL. The first formatted binary line after the CIL line must be COMD section #4 or COMD section #1. After each core image is loaded, SYSLOD is set to load a new core image. If the beginning of the new core image is not COMD section #4 or #1, this error message is issued.

COMD SECTION #4 SEQUENCE ERROR

This error message occurs if the LICIL is being loaded from paper tapes, and one or more tapes are out of order.

INPUT ERROR

After a READ, the status in the buffer header has indicated that one of the following errors occurred:

a) invalid line error

b) checksum error

c) character parity error or illegal binary format

d) device parity error

LOGICAL BLOCK SIZE ERROR

This error message is issued when the logical block size specified for the NS switch is not an integral multiple of the physical block size for the disk.

END OF DISK BEFORE CIL COMPLETE

The last block number of the output disk has been written, but the CIL is not complete.

ILLEGAL EMT CALL

An EMT call was made that was not recognizable as a valid DOS/BATCH EMT. Notify your Software Support representative.

FATAL ERROR RETRY

A fatal error has aborted the current operation. It must be retried.

MT DISASTER NXM or ILC - IRRECOVERABLE

An irrecoverable I/O error has occurred while reading or writing magtape.

IRRECOVERABLE MT PROBLEM

A persistent error has occurred while reading magtape.

LICIL FILE NOT FOUND

The specified LICIL could not be found under UIC [1,1] or UIC [2ØØ,2ØØ].

NO SPACE FOR CIL

Not enough contiguous disk space is available to create the CIL. Either (1) too many files already exist on the disk, or (2) there are too many bad blocks on the disk.

BLOCK Ø OR BLOCK 1 BAD

Either block Ø or block 1 on the disk is bad; the system cannot be generated, as both these blocks are essential to the disk file structure.

PACK IS TOO BAD

The current disk pack cannot be used, as there are too many bad blocks (BADB.SYS is full).

ESSENTIAL DISK BLOCK HAD I/O ERROR

A MFD, UFD, or bit map block could not be written without encountering I/O errors. Note that these blocks are written after the verification phase.

C.3 NOTES

If the default block size is not used, the block size must be an integral multiple of the default size for the disk.

The COMD (COMmunication Directory) contains a code number that identifies the kind of information that follows. SYSLOD expects the first formatted binary line it receives on input to be identified as code #4 (indicating that it is a LICIL).

On occassion, certain hardware problems such as magtape and RK disk head alignment, UNIBUS time out traps, and disk controller errors cause SYSLOD to halt. The system manager should immediately contact a DIGITAL field service representative.

APPENDIX D

SUMMARY OF DOS-11 ERROR MESSAGES

Following is a complete summary of all error messages which can appear when using the DOS Monitor and system programs.

D.1 ACTION MESSAGES

Error Code	Additional Information	Meaning
A001	User Call Address	Disk Address Error
A002	Device Name (RAD50)	Device not ready
A003	Link Block Address	No device or illegal device or INIT
A004	User Call Address	DECtape error; command CO will retry the operation
A005	PAUSE Number	FORTRAN PAUSE
A006	Irrelevant	Loading paper tape out of order on Pass 2 of Linker. Load specified module next.
A007	Call Address	MAGtape opened file exists
A010	0	MAGtape read error during open
A043	Disk Pack Block Number	This is the block that is bad

D.2 INFORMATIONAL MESSAGES

Error Code	Additional Information	Meaning
I350	STOP Number	FORTRAN STOP
I351	Err Max Cnt Reached	FORTRAN abnormal exit
I352	Addr of Log. Device	No FORTRAN logical device
I353	Class, error #	No logging device
I354	0	Disk was not zeroed
I354		Illegal response to CONFIRM: (the DK11 Disk cartridge was not zeroed)

D.3 FATAL MESSAGES

Error Code	Additional Information	Meaning
F000	Request Address	Dataset not INITed
F001	Request Address	Stack overflow
F002	Request Address	Invalid call
F003	Request Address	Invalid .TRAN function
F005	Request Address	.RLSE error (no .CLOSE after .OPEN on file-structured device)
F006	Request Address	Device full (new file cannot be .OPENed)
F007	Request Address	No buffer space
F010	Request Address	Illegal .READ/.WRITE (incorrect mode for device or file not OPENed correctly)
F011	Request Address	Illegal open (unused code or type unsuitable for device)
F012	Request Address	Invalid open (no program return provided for failure)
F014	Request Address	Bit map failure (device error on trying to read bit map)
F015	Request Address	DECtape error (nonexistent memory addressed or end-zone reached dummy transfer)
F016	Block Number	DECtape search failure (block requested cannot be found)
F017	Device (RAD50)	File structures device parity error
F020	Irrelevant	Too many datasets using low-speed paper tape (a maximum of one for each direction is allowed)
F021	Irrelevant	Program Loader read failure
F022	Irrelevant	Program Loader format error (program is not in absolute load format)
F023	Program Size	Program too large for core available

Error Code	Additional Information	Meaning
F024	Request Address	Illegal file structures operation, e.g., Delete or Rename of an open file.
F025	Device	Master Directory full (when attempting to add UIC)
F026	Disk Control Status Register	Disk transfer failure (hardware error or persistent parity failure)
F027	Status	RK11 hardware failure
F030	Error Class/Number	FORTRAN system error
F031	Address of Logical Device	End of file (FORTRAN)
F032	Status (MT)	Magtape hardware failure
F033	Call Address	Invalid special function
F034	Call Address	Invalid conversion code
F035	Block Number	Disk (DK) failure
F036	Lowest slot used by Tasks	RSX Loader-no slot available
F037	Lowest slot used by Tasks	RSX Loader-illegal slot specified
F040	Low address of Task Code	RSX Loader-ABS Task does not fit
F041	Low Address of Stack	RSX Loader-ABS Task, not enough room for Stack
F042	Error Register	RP Hardware failure
F043	Block Number	RP Failure
F240	Irrelevant	No space for file allocate
F342	Contents of PC	Error Trap
F344		Reserved instruction trap
F346		Trace trap
F350		Power Fail trap
F352		Trap Instruction trap
F356		Unexpected device interrupt

D.4 SYSTEM PROGRAM MESSAGES

Error Code	Additional Information	Meaning
S200	0	Too many .CSECT directives
S201	0	Conditionals nested too deeply
S202	Error Status Byte	EOD or device error on .WRITE or .READ
S203	0	Illegal switch, or too many switches, or illegal switch value, or switch value not given
S204	0	Too many or too few output files
S205	0	Too many or too few input files
S206	0	Input file not specified in command string
S207	Error Status Byte	EOD or device on .TRAN
S210	0	Unrecognized symbol table entry
S211	0	An RLD references a global name which cannot be found in the symbol table
S212	0	An RLD contains a location counter modification command which is not last
S213	0	Object module does not start with a GSD
S214	0	The first entry in the module is not the module name
S215	0	An RLD references a section name which cannot be found
S216	0	The TRA specification references a nonexistent module name
S217	0	The TRA specification references a nonexistent section name
S220	0	An internal jump table index is out of range
S221		Unassigned
S222		Unassigned

Error Code	Additional Information	Meaning
S223	0	No more room for CSI input buffer or Monitor's file manager routine, or Monitor's Library search buffer
S224		Unassigned
S225	0	Program too large or top too low (program has been linked below zero in memory)
S226	0	An open angle bracket, <, is present in a line other than the first
S227	1:No Primary Output 2:Sec In = Sec Out 3:Sec In = Pri Out 4:Pri In = Sec Out 5:Pri In = Sec In 6:Pri Out = Sec Out	Illegal file combination; arguments 2-6 for Editor-type commands Pri = Primary File Sec = Secondary File
S230	Error Status Byte	Error on .BLOCK I/O
S231		Illegal command, file-structured device required
S232		No more than one action switch permitted
S233		Specified UIC not found in MFD
S234		Null file name given where file name required
S235		No files found in UFD
S236		Operation applicable to DECtape only
S237		File not found during file recovery operation
S240		No space for file allocate
S241		MFD is full
S242		Meaningless command; no action taken
S243	0	No < in first line of command
S244	0	Already past requested position
S245	0	Object module not found, could be out of order
S246	0	Illegal library format

Error Code	Additional Information	Meaning
S247	0	Listing requested, but unable to read output library from specified output device
S250	0	Core library symbol table not specified first or consecutively
S251	0	No files found for *request
S252	0	File name given when none allowed
S253	0	Linker error
S254	0	It is illegal to ZEro the system resident disk

D.5 WARNING MESSAGES

Error Code	Additional Information	Meaning
W002	RAD50 NAME	Device time out
W043	Block Number	Transfer error while using ITRAN to zero the Disk
W300	0	Non-unique module name
W301	0	Byte relocation error
W302	0	Multiple definitions of global symbol
W303	0	Buffer overflow
W304	0	Macro too long
X305	0	Recursive macro not allowed
W306	0	Empty save buffer
W307	0	Search failure
W310	0	No room to UNSAVE
W311	0	End of data
W312	0	Line feed illegal
W313	0	Negative argument illegal
W314	0	No arguments allowed
W315	0	Illegal argument
W316	0	Illegal text string
W317	0	Illegal command
W320	0	Page buffer almost full

Error Code	Additional Information	Meaning
W321	0	Attempting to write to closed file
W322	0	Undefined Global Symbols
W323	0	Illegal size of named .CSECT, or illegal entry in named .CSECT, or Tasks's named .CSECT size too large (RSX)
W324	0	Too many entries in Task's named .CSECT (RSX)
W325	0	Illegal Priority Specification in Real-Time Header (RSX)
W350	Number of Failures	RSX-Power Fail

APPENDIX E

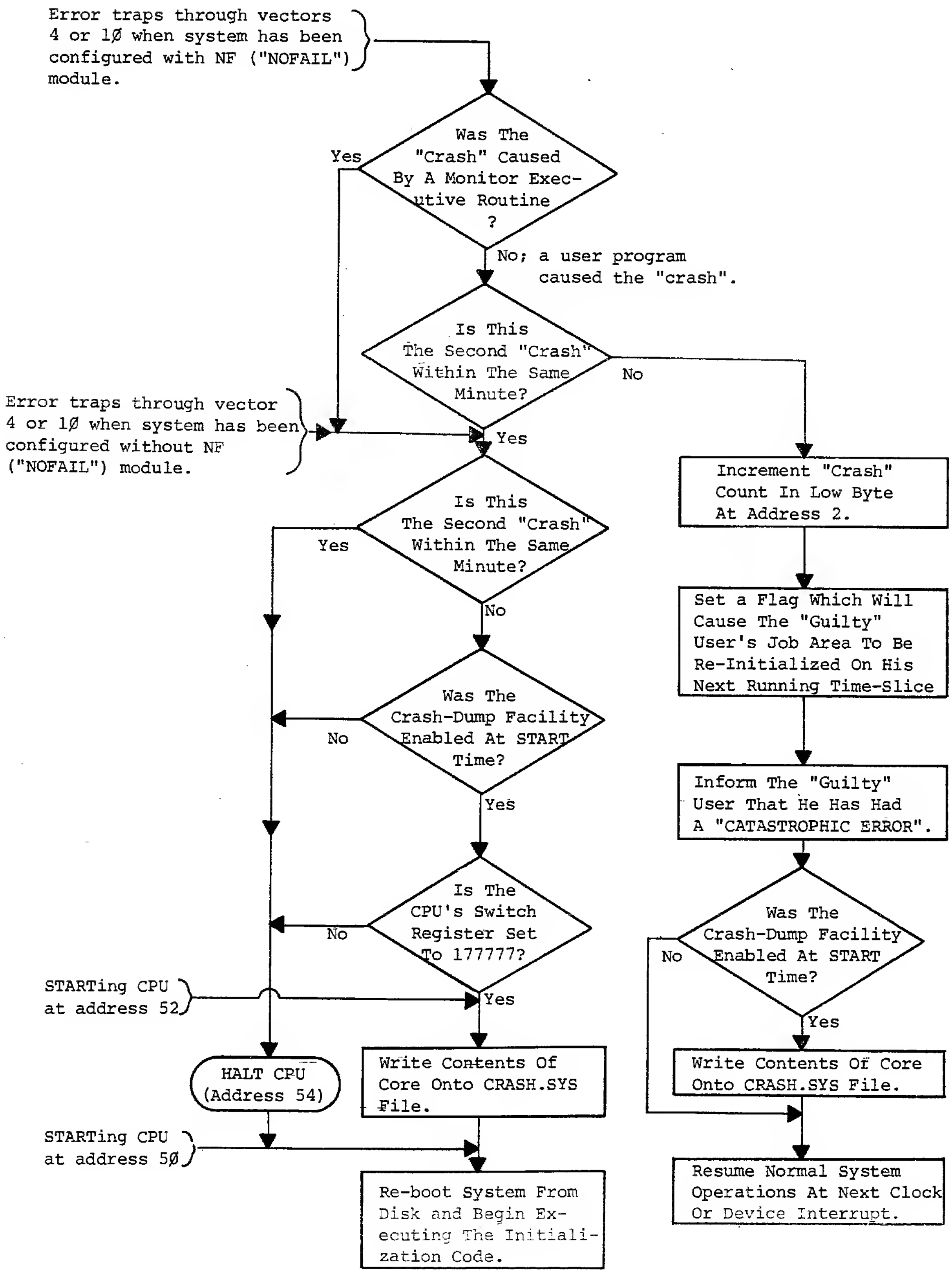
RSTS INITIALIZATION ERROR MESSAGES

If the initialization code detects an error, it prints a message of the form FATAL INITIALIZATION ERROR AT nnnnnn where nnnnnn is a relative relocatable address in the related module. The following addresses and related conditions describe the errors.

ØØ3112'	NO USABLE USER SPACE
ØØ4724'	ERROR DURING FIP OPERATION
ØØ4756'	?OVER.SYS[Ø,1]
ØØ476Ø'	?ERRM.SYS[Ø,1]
ØØ4762'	?SWAP.SYS[Ø,1]
ØØ546Ø'	?BUFF.SYS[Ø,1]
ØØ6376'	MFD CREATE FAILURE
ØØ64ØØ'	ACCT [1,1] SEARCH FAILURE
ØØ6564'	MISMATCHED CLUSTERS
ØØ7Ø74'	CANNOT FIND CREATED FILE
ØØ7334'	FILE NOT FOUND FOR SHIFTING
ØØ7562'	CANNOT FIND FILE FOR DUMP
Ø1Ø526'	FILE NOT FOUND IN 'OLD' OF REFRESH
Ø11334'	?BADB.SYS[Ø,1]
Ø12276'	SMALL BUFFER POOL FULL
Ø123ØØ'	ERROR RETURN FROM FILE PROCESSOR
Ø14774'	BAD LISTHT FOR FILE
Ø15746'	UNRECOGNIZED SYSTEM DISK

APPENDIX F

RSTS-11 AUTO-RECOVERY AND AUTO-RESTART FLOWCHART



APPENDIX G

DECPACK DISTRIBUTION

G.1 RSTS-11 DECpack Kit and System Generation Procedures

The RSTS-11 DECpack software is distributed on two RKØ3/Ø5 DECpack disk cartridges. The contents of the supplied Generation DECpack are the same as the contents of DECtapes #Ø, #1, and the DOS Monitor of the RSTS-11 DECtape kit, except that the DOS Monitor is VØ8-Ø2. The contents of the Library DECpack correspond identically to the contents of DECtape #2. (Refer to section 2.1 for a description of the contents of the DECtapes.)

The generation of the RSTS-11 system using the DECpack distribution entails the following steps. The system manager should read the entire guide before attempting to follow these steps, which are described in sections G.2 through G.8.

1. Activate the supplied DOS-11 VØ8-Ø2 Monitor.
2. Format two new DECpack cartridges using the DSKINT stand-alone program.
3. Using the ROLLIN stand-alone program, copy the supplied Generation DECpack to one of the newly formatted cartridges. The original distributed Generation DECpack is then stored in a safe place.
4. Run the SYSGEN program, answer the SYSGEN questions, and follow the system generation instructions printed in the dialogue preview.
5. After building the RSTS-11 CIL on the system disk, follow the REFRESH initialization procedures.
6. The final step consists of building the system library using the supplied Library DECpack and establishing the system accounts, text messages, and control files.
7. Copy the newly-created RSTS-11 system onto a backup cartridge.

G.2 Booting the DOS Monitor in Core

Mount the RSTS V4A-12 Generation DECpack (DEC-11-ORDPA-A-HC) on the RKØ3/Ø5 drive unit Ø. Write enable drive unit Ø. Bootstrap the Monitor by doing the following:

Move the CPU's ENABLE/HALT switch to its HALT position.

If the BM792-YB Hardware Loader is on the system:

Set the CPU Switch Register to 173100.
Depress the CPU LOAD ADDRESS switch.
Move the CPU ENABLE/HALT switch to its ENABLE position.
Set the CPU Switch Register to 177406.
Depress the CPU START switch.

If the MR11-DB Hardware Loader is on the system:

Set the CPU Switch Register to 173110.
Depress the CPU LOAD ADDRESS switch.
Move the CPU ENABLE/HALT switch to its ENABLE position.
Depress the CPU START switch.

When the Monitor is loaded, it prints its header identification, followed on another line by a $ character which indicates that the system is ready to accept command strings. The printout appears as follows:

```
DOS V08-02
$
```

If the message does not appear, reset the CPU ENABLE/HALT switch to HALT and try again, checking carefully the values loaded into the registers.

G.3 Formatting the Cartridges

Mount one of two new DECpack cartridges on RK03/05 drive unit 1. Run CILUS and boot in the disk initializer program DSKINT by following the dialogue shown below:

```
$RUN CILUS
CILUS V01A69
#DSKINT.CIL[1,1]/BOOT
```

NOTE

Messages printed by the system are indicated by underlining. Responses to be entered by the user must be terminated by pressing the RETURN key. The RETURN key is echoed by the performance of a line-feed/carriage return operation.

When the DSKINT program is loaded, the following header and query lines are printed.

```
RSTS DISK INITIALIZER V4A-ØØ
DISK (RF, RK, RP) =
```

At this point, remove the DECpack distribution cartridge from RK03/05 drive unit Ø and replace it with the second new DECpack cartridge. Consult section 6.2 of this manual for information on using the DSKINT program. The dialogue appears as follows:

```
DISK (RF,RK,RP) = RK
DRIVE NUM (RK,RP) OR NUM PLATTERS-1 (RF) =1
DISK ID (UP TO 6 CHARS) =RSTS11
PACK CLUSTER SIZE (1,2,4,8 OR 16) =1
MFD PASSWORD (UP TO 6 CHARS) =VØ4A12
MFD CLUSTER SIZE (AT LEAST=PACK CLUST) =1
PUBLIC (PUB) OR PRIVATE (PRI) =PUB
PACK FORMATTING (Y OR N) =Y
READY DISK THEN TYPE CR
INITIALIZATION COMPLETE

RSTS DISK INITIALIZER V4A-ØØ

DISK (RF,RK,RP) =
```

When the initialization and formatting is completed, the program is reinitialized and prints the header and first query line.

Repeat the same procedures for the second cartridge, but use DRIVE NUM = Ø in the second question. Leave the first cartridge mounted when the initialization is completed, and replace the second cartridge in drive unit Ø with the DECpack distribution cartridge, DEC-11-ORDPA-A-HC.

G.4 Copying the Distribution DECpack

Boot the DOS Monitor into core again as described in section G.2. Run CILUS, boot ROLLIN into core and copy the supplied DECpack, which is mounted on drive unit Ø, onto the newly formatted cartridge which remains mounted on drive unit 1. The dialogue is as follows:

```
DOS VØ8-Ø2
$RUN CILUS
CILUS VØ1A69
#ROLLIN.CIL [1,1]/BOOT

ROLLIN VØ5A
#DK1:<DKØ:
#
```

WARNING

> Be sure that the copy command is typed EXACTLY as shown before typing the RETURN key. Do NOT under any circumstances type a ">" character or type the unit numbers in reverse order.

The completion of the copy operation is signalled by the printing of the # character. Upon completion of the copy operation, dismount the supplied DECpack from drive unit Ø and store it in a safe place. Transfer the copy from drive unit 1 to drive unit Ø. Write Enable unit Ø.

G.5 Running the SYSGEN Program

With the copy of the supplied DECpack mounted on unit Ø, boot the DOS Monitor into core as shown, perform the login procedure, and run the SYSGEN program.

```
#/BOOT:DK
DOS VØ8-Ø2

$DATE dd-mon-yy              [for example: 13-NOV-72]
$TIME hh:mm                  [for example: 13:ØØ]
$LOGIN 1,1
DATE:-13-NOV-72
TIME:-13:ØØ:Ø5
$RUN SYSGEN

RSTS-11 V4A SYSTEM GENERATION
```

The date and time must be given in the format shown. When the SYSGEN program header is printed, answer the questions printed by SYSGEN. Refer to sections 2.2 and 2.3 of this guide for the complete explanation of the process. When all the questions are answered, the SYSGEN program prints (on the line printer or on the console Teletype, as directed) a preview image of the subsequent system generation dialogue. Follow the instructions and use the exact command strings which SYSGEN has printed in the preview. When the last command indicated by the SYSGEN preview is executed, the RSTS-11 CIL exists on the DECpack on drive unit 1. Dismount the cartridge from unit Ø and transfer the cartridge just built on drive unit 1 to drive unit Ø. Proceed to "Refreshing the System Disk" below.

G.6 Refreshing the System Disk

Boot the newly-created RSTS-11 system into core by following the procedures described in section G.2. Next, refresh the system disk following the procedures described in sections 2.2.11 and 2.3.15. Perform the initialization as described in section 2.3.16.

G.7 Building the System Library

When the RSTS-11 Monitor has been initialized and prints the READY message, it will be necessary to mount the distributed Library DECpack and to build the system library by retrieving files from the Library DECpack and following procedures similar to those outlined in sections 2.3.17 through 2.3.21.

G.7.1 Mounting the Library DECpack

After the system prints the READY message, mount the RSTS V4A-12 Library DECpack (DEC-11-ORPTA-A-HA) on RKØ3/Ø5 drive unit 1. Write enable drive unit 1. Type the following immediate mode statements:

```
READY
X$ = SYS( CHR$(6) + CHR$(-1Ø) + "DK1:V4ALIB" )
READY
Y$ = RIGHT( X$ , 4 )
READY
Z$ = SYS( CHR$(6) + CHR$(3) + CHR$(Ø) + Y$ )
READY
```

The distribution Library DECpack will then be logically mounted, and files on it may be accessed.

G.7.2 Retrieving the Library Files

When the Library DECpack has been logically mounted, as outlined above, follow the instructions in sections 2.3.17 through 2.3.21. Substitute DK1: for the device designator DTØ: shown in the example command strings, since the system programs and text files exist on DECpack drive unit 1 rather than DECtape. Sample command lines are shown below:

```
READY
OLD DK1:LOGIN
READY
COMPILE
READY
OLD DK1:LOGOUT
READY
etc...
```

When the system library has been built, the distributed Library DECpack may be removed by following instructions contained in section 5.3.2.1.

G.8 Preserving the RSTS-11 System

When the RSTS-11 system has been generated, use the ROLLIN stand-alone program as described in section 6.1 to create a copy of the system on the other new DECpack cartridge.

READER'S COMMENTS

Digital Equipment Corporation maintains o continuous effort to improve the quality ond usefulness of its publications. To da this effectively we need user feedback -- your criticol evoluotion of this monuol.

Please camment an this manuol's completeness, accurocy, orgonization, usability and read-obility.

Did you find errors in this manual? If sa, specify by page.

How can this manual be improved?

Other comments?

Please state your pasition. ______ Date: ______

Nome: ______ Orgonizotion: ______

Street: ______ Department: ______

City: ______ State: ______ Zip or Country ______

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Information Services
Programming Department
Maynard, Massachusetts 01754

HOW TO OBTAIN SOFTWARE INFORMATION

SOFTWARE NEWSLETTERS, MAILING LIST

The Software Communications Group, located at corporate headquarters in Maynard, publishes software newsletters for the various DIGITAL products. Newsletters are published monthly, and keep the user informed about customer software problems and solutions, new software products, documentation corrections, as well as programming notes and techniques.

There are two similar levels of service:

- The Software Dispatch
- The Digital Software News

The Software Dispatch is part of the Software Maintenance Service. This service applies to the following software products:

PDP-9/15
RSX-11D
DOS/BATCH
RSTS/E
DECsystem-10

A Digital Software News for the PDP-11 and a Digital Software News for the PDP-8/12 are available to any customer who has purchased PDP-11 or PDP-8/12 software.

A collection of existing problems and solutions for a given software system is published periodically. A customer receives this publication with his initial software kit with the delivery of his system. This collection would be either a Software Dispatch Review or Software Performance Summary depending on the system ordered.

A mailing list of users who receive software newsletters is also maintained by Software Communications. Users must sign-up for the newsletter they desire. This can be done by either completing the form supplied with the Review or Summary or by writing to:

Software Communications
P.O. Box F
Maynard, Massachusetts 01754

SOFTWARE PROBLEMS

Questions or problems relating to DIGITAL's software should be reported as follows:

North and South American Submitters:

Upon completion of Software Performance Report (SPR) form remove last copy and send remainder to:

Software Communications
P.O. Box F
Maynard, Massachusetts 01754

The acknowledgement copy will be returned along with a blank SPR form upon receipt. The acknowledgement will contain a DIGITAL assigned SPR number. The SPR number or the preprinted number should be referenced in any future correspondence. Additional SPR forms may be obtained from the above address.

All International Submitters:

Upon completion of the SPR form, reserve the last copy and send the remainder to the SPR Center in the nearest DIGITAL office. SPR forms are also available from our SPR Centers.

PROGRAMS AND MANUALS

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center.

Digital Equipment Corporation Software Distribution Center 146 Main Street Maynard, Massachusetts 01754	Digital Equipment Corporation Software Distribution Center 1400 Terra Bella Mountain View, California 94043

Outside of the United States, orders should be directed to the nearest Digital Field Sales Office or representative.

USERS SOCIETY

DECUS, Digital Equipment Computers Users Society, maintains a user exchange center for user-written programs and technical application information. The Library contains approximately 1,900 programs for all DIGITAL computer lines. Executive routines, editors, debuggers, special functions, games, maintenance and various other classes of programs are available.

DECUS Program Library Catalogs are routinely updated and contain lists and abstracts of all programs according to computer line:

- PDP-8, FOCAL-8, BASIC-8, PDP-12
- PDP-7/9, 9, 15
- PDP-11, RSTS-11
- PDP-6/10, 10

Forms and information on acquiring and submitting programs to the DECUS Library may be obtained from the DECUS office.

In addition to the catalogs, DECUS also publishes the following:

DECUSCOPE	-The Society's technical newsletter, published bi-monthly, aimed at facilitating the interchange of technical information among users of DIGITAL computers and at disseminating news items concerning the Society. Circulation reached 19,000 in May, 1974.
PROCEEDINGS OF THE DIGITAL EQUIPMENT USERS SOCIETY	-Contains technical papers presented at DECUS Symposia held twice a year in the United States, once a year in Europe, Australia, and Canada.
MINUTES OF THE DECsystem-10 SESSIONS	-A report of the DECsystem-10 sessions held at the two United States DECUS Symposia.
COPY-N-Mail	-A monthly mailed communique among DECsystem-10 users.
LUG/SIG	-Mailing of Local User Group (LUG) and Special Interest Group (SIG) communique, aimed at providing closer communication among users of a specific product or application.

Further information on the DECUS Library, publications, and other DECUS activities is available from the DECUS offices listed below:

DECUS Digital Equipment Corporation 146 Main Street Maynard, Massachusetts 01754	DECUS EUROPE Digital Equipment Corp. International (Europe) P.O. Box 340 1211 Geneva 26 Switzerland

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 and PDP-12
Digital Software News for the PDP-11
Digital Software News for 18-bit Computers

These newsletters contain information applicable to software available from Digital's Software Distribution Center. Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contract at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problems to:

Digital Equipment Corporation
Software Information Service
Programming Department
Maynard, Massachusetts 01754

These forms, which are provided in the software kit, should be fully filled out and accompanied by Teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. USA customers may order directly from the Software Distribution Center in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information, please write to:

Digital Equipment Corporation
DECUS
Programming Department
Maynard, Massachusetts 01754

digital

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS 01754